

Development of Authentication-
based CAPTCHA Mechanism on Touch Screen Environment



A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Science and Information Technology
Department of Mathematics and Computer Science
FACULTY OF SCIENCE
Chulalongkorn University
Academic Year 2020
Copyright of Chulalongkorn University

การพัฒนากลไกการระบุตัวตนด้วยแคปซันบนพื้นฐานสภาพแวดล้อมของจอภาพแบบสัมผัส



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการ

คอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2563

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	Development of Authentication- based CAPTCHA Mechanism on Touch Screen Environm ent
By	Miss Nilobon Nanglae
Field of Study	Computer Science and Information Technology
Thesis Advisor	Associate Professor PATTARASINEE BHATTARAKOSOL, Ph.D.

Accepted by the FACULTY OF SCIENCE, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

..... Dean of the FACULTY OF SCIENCE
(Professor POLKIT SANGVANICH, Ph.D.)

DISSERTATION COMMITTEE

..... Chairman
(Associate Professor Panjai Tantatsanawong, Ph.D.)

..... Thesis Advisor
(Associate Professor PATTARASINEE BHATTARAKOSOL,
Ph.D.)

..... Examiner
(Associate Professor Nagul Cooharojananone, Ph.D.)

..... Examiner
(Associate Professor Thiti Bovornratanaraks)

..... External Examiner
(Assistant Professor Kanokwan Atchariyachanvanich,
Ph.D.)

นิโลบล นางแล : การพัฒนากลไกการระบุตัวตนด้วยแคปช่าบนพื้นฐานสภาพแวดล้อมของจอภาพแบบสัมผัส. (Development of Authentication-based CAPTCHA Mechanism on Touch Screen Environment) อ.ที่ปรึกษาหลัก
: ภัทรสินี ภัทรโกศล

แคปต์ช่าเป็นการทดสอบความปลอดภัยอย่างง่ายที่นำมาใช้เพื่อแยกแยะระหว่างมนุษย์และโปรแกรมอัตโนมัติมานานหลายทศวรรษ ซึ่งแคปต์ช่าถูกใช้กันอย่างแพร่หลายในเว็บไซต์เชิงพาณิชย์ เช่น บริการอีเมล และ เว็บไซต์เครือข่ายสังคมออนไลน์ เพื่อปกป้องระบบจากการโจมตีของซอฟต์แวร์อัตโนมัติ อย่างไรก็ตามมีการคิดค้นเทคนิคต่าง ๆ เพื่อโจมตีแคปต์ช่า ซึ่งหนึ่งในเทคนิคเหล่านี้คือการโจมตีของบุคคลที่สามที่ถูกจ้างวานเพื่อโจมตีแคปต์ช่า ดังจุดประสงค์ของแคปต์ช่าซึ่งถูกออกแบบเพื่อแยกความแตกต่างระหว่างมนุษย์กับโปรแกรมอัตโนมัติ ทำให้แคปต์ช่าไม่สามารถแยกความแตกต่างระหว่างผู้ใช้ที่เป็นมนุษย์และผู้โจมตีที่เป็นมนุษย์นอกกฎหมายได้ ดังนั้นงานวิจัยนี้จึงเสนอแคปต์ช่ารูปแบบใหม่ ที่ถูกสร้างขึ้นสำหรับแต่ละบุคคลสำหรับผู้ใช้งานระบบ เทคนิคที่น่าเสนอนี้จะผสมผสานระหว่างไบโอเมตริกและโพรไฟล์ของผู้ใช้งานระบบเพื่อนำมาสร้างแคปต์ช่าที่เหมาะสมที่สุดสำหรับแต่ละบุคคล ซึ่งผู้บุกรุกไม่สามารถโจมตีได้ง่าย แม้แต่มนุษย์ที่เป็นผู้บุกรุกก็ไม่สามารถผ่านได้ นอกจากนี้แคปต์ช่าที่เสนอนี้สามารถใช้เป็นรหัสผ่านชั่วคราวสำหรับผู้ใช้งานทุกคน และทุกครั้งที่จะเข้าสู่ระบบจะมีการสุ่มแคปต์ช่าและไม่ซ้ำกัน ส่วนการประเมินประสิทธิภาพของเทคนิคที่น่าเสนอนี้พบว่า หากผู้ใช้ทราบ CAPTCHA แบบเต็ม ระบบสามารถระบุผู้ใช้ที่แท้จริงได้ด้วยความแม่นยำ 100% แต่สำหรับผู้บุกรุกจะมีการระบุผู้บุกรุกเพียง 51.0% ราวกับว่าพวกเขาเป็นผู้ใช้ที่แท้จริง อย่างไรก็ตามการโจมตีของโปรแกรมอัตโนมัติจะต้องใช้เวลาในการแก้ปัญหาเป็นเวลานานมากและพยายามที่ล้มเหลวมาก ซึ่งในการทำงานจริงอาจถูกขัดจังหวะด้วยเวลาที่จำกัดของระบบ ดังนั้นผู้บุกรุกทั้งหมดจึงไม่สามารถเข้าถึงได้ตามต้องการ

สาขาวิชา วิทยาการคอมพิวเตอร์และ เทคโนโลยีสารสนเทศ
ลายมือชื่อนิสิต

ปีการศึกษา 2563
ลายมือชื่อ อ.ที่ปรึกษาหลัก

5773103023 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORD: Text-based CAPTCHA, Keystroke dynamics, Biometric Authentication,
Mobile device, Personal Identity

Nilobon Nanglae : Development of Authentication-
based CAPTCHA Mechanism on Touch Screen Environment. Advisor: Assoc.
Prof. PATTARASINEE BHATTARAKOSOL, Ph.D.

CAPTCHA is a simple security test that was introduced to distinguish among humans and bots for decades. CAPTCHAs have been widely used on commercial sites, such as email service, and social networking sites, for protecting the system from automated software attackers. However, various techniques have been invented to break CAPTCHA, and one of these techniques is the 3rd party attacks. So, the design of CAPTCHA is unable to distinguish between human users and illegitimate human attackers. Thus, this research proposed a new type of CAPTCHA that is individually generated for an individual user. The proposed technique merges between biometrics and the user's profile to obtain the most suitable CAPTCHA that cannot easily be broken by all intruders, even the human in the CAPTCHA farm. Besides, this proposed CAPTCHA can be used as a temporary password for every user, every login time, because of its randomness and uniqueness. The performance evaluation of this proposed technique indicates that if a user knows the full CAPTCHA, the system can determine the true user with 100% accuracy, but for the intruders, only 51.0% of the intruders would be identified as if they were the true user. Nonetheless, the bots attack must spend a very long-time solving and more failed attempts, which, in real-life working, it could be interrupted by the time limit of the system. Therefore, all bots cannot gain access as required.

Field of Study: Computer Science and
Information Technology

Student's Signature

Academic Year: 2020

Advisor's Signature

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my research supervisor, Associate Professor Dr. Pattarasinee Bpattarakosol, Ph.D., for giving me the opportunity to do research and providing invaluable guidance throughout this research. Her dynamism, vision, sincerity and motivation have deeply inspired me. She has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under her guidance. I am extremely grateful for what she has offered me. I would also like to thank her for her empathy, and great sense of humor. I am extending my heartfelt thanks to her family for their support me during work on my research and thesis preparation.

Beside my advisor, I would also like to thank Assoc. Prof. Panjai Tantatsanawong, Ph.D., Assoc. Prof. Nagul Cooharajanone, Ph.D., Assoc. Prof. Thiti Bovornratanaraks, Ph.D., and Asst. Prof. Kanokwan Atcharyachanvanich, Ph.D., my committee, for their time and assistance.

I am extremely grateful to my family for their love, prayers, caring and sacrifices for educating and preparing me for my future. My Special thanks goes to my friend, Wittakorn Keeratichayakorn, for recommendation about the coding, and continuing support to complete this research work.

I would like to say thanks to all my friends who have participate in my research as a subject in the experiment. Special thanks to my friend from IPST colleagues for encouragement and advice. Also, I express my thanks to INSET Laboratory colleagues for his friendship, empathy, and great sense of humor.

The 90th Anniversary of Chulalongkorn University, Rachadapisek Sompote Fund, and The 100th Anniversary Chulalongkorn University Fund for Doctoral

Scholarships is gratefully acknowledged for financial support of this research

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Nilobon Nanglae



TABLE OF CONTENTS

	Page
.....	iii
ABSTRACT (THAI).....	iii
.....	iv
ABSTRACT (ENGLISH).....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
Chapter 1 Introduction.....	1
1.1 Background and Importance.....	1
1.2 Objective.....	4
1.3 Scope of the research.....	5
1.4 The expected outcomes.....	5
1.5 Definition.....	6
1.6 The structure of thesis.....	7
Chapter 2 Literature Review.....	9
2.1 CAPTCHAs.....	9
2.2 Overview of Breaking.....	11
2.2.1 Breaking Traditional Attackers.....	11
2.2.2 Overview of Breaking Methods towards Text-based CAPTCHA.....	15
2.3 Keystroke Dynamics.....	19

Chapter 3 Methodology.....	22
3.1 Preliminary Test.....	22
3.1.1 Program for finding differences between right-handed and left- handed	22
3.1.2 Program for collecting the typing rhythm	24
3.2 Feature Analysis.....	26
3.2.1 Data Collection.....	26
3.2.2 Preprocessing.....	28
3.2.3 Feature Extraction.....	31
3.2.4 Classification and Evaluation.....	36
3.3 The Experimental of the Proposed Text-based CAPTCHA	37
3.3.2 User Verification	45
3.3.3 Performance evaluation	50
Chapter 4 Results	59
4.1 Results of factor analysis.....	59
4.1.1 Mann-Whitney U test: Test for Gender’s effect	60
4.1.2 Kruskal-Wallis test: Test for Fingertip’s effect	60
4.1.3 Factors Suitability Testing	61
4.1.4 Factor weight	66
4.2 The consideration of typing characteristics in various conditions	68
4.3 The result of the first evaluation.....	76
4.4 The result of the second evaluation	77
4.4.1 Brute force online simulator	77
4.4.2 Typing CAPTCHA Simulator Bot	78
4.4.3 Typing CAPTCHA with Simulator Bot including special conditions.....	78

Chapter 5 Discussion.....	80
Chapter 6 Conclusion.....	85
Chapter 7 Future Work.....	86
Appendix	87
REFERENCES	94
VITA.....	103



LIST OF TABLES

	Page
Table 1 List of CAPTCHAs that can protect the 3 rd party attack.....	18
Table 2 The related studies on keystroke dynamics based on touchscreen device .	20
Table 3 Results of Independent Sample T-test.....	23
Table 4 The formula for calculating keystroke values	40
Table 5 Samples of time comparison between the input time with the valid time domain.....	49
Table 6 Averages of all metrics.....	65
Table 7 The weight factors of 8 factors	66
Table 8 The result of the real user tries to enter the Text-based CAPTCHA as “r t h r o s” and “e t t o a t r”	76
Table 9 The result of the other user tries to enter the Text-based CAPTCHA as “r t h r o s” and “e t t o a t r”	77
Table 10 The result of the brute force online simulator to guess 6 characters length and 7 characters length	78
Table 11 The result of the simulator bot with special condition to crack 6 characters length and 7 characters length.....	79
Table 12 Performance comparison among authentication classification factors.....	81

LIST OF FIGURES

	Page
Figure 1 Collecting screen and text.....	22
Figure 2 The sample of the Profile page.....	24
Figure 3 Sentence Entering page.....	25
Figure 4 Differentiation line graph of all samples	25
Figure 5 Steps of all 5 executing procedures in this experiment.....	26
Figure 6 Data collection process.....	27
Figure 7 The Database Diagram on the Cloud	28
Figure 8 The acceptant interval of the valid data set.....	29
Figure 9 Demonstrations of the keystroke features	30
Figure 10 System Architecture for Authentication Classification.....	37
Figure 11 Forms to collect data.....	39
Figure 12 Example of Keystroke features calculation.....	41
Figure 13 Example of the streaming transaction from a client to the CAPTCHA_Key database.....	42
Figure 14 The area under 2 standard deviations using the Empirical rule.....	43
Figure 15 Samples of 6 digits of the Text-based CAPTCHA generated by the CAPTCHA Generator.....	47
Figure 16 A simple example of visualization of Gradient Boosted Trees.....	50
Figure 17 The evaluation process in the first phase.....	52
Figure 18 The accuracy values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.....	62

Figure 19 The precision values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.....	62
Figure 20 The recall values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.....	63
Figure 21 The different execution times using Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.....	64
Figure 22 The accuracy that cutting the factors.....	67
Figure 23 comparison of the individual typing character between a legitimate user and intruders	68
Figure 24 Display all typing times of di-graph characters of a legitimate user.....	69
Figure 25 The line graph of dwell times based on an individual Text-based CAPTCHA typing of a legitimate user.....	70
Figure 26 The line graph of dwell and digraph times based on an individual Text-based CAPTCHA typing of a legitimate user.....	71
Figure 27 Legitimate user types individual Text-based CAPTCHA that include di-graph character (Digraph only).....	72
Figure 28 The line graph of dwell times based on a derived Text-based CAPTCHA typing by an intruder.....	73
Figure 29 The line graph of dwell times for each character of an individual Text-based CAPTCHA typing by a intruder.....	74
Figure 30 Intruder types individual Text-based CAPTCHA with digraph characters (Digraph only).....	75

Chapter 1

Introduction

This chapter consists of information about background and importance of this research. That is including objective, scope of this study, its constraints, and expected outcome of this experimental. To make this clear for reader to understand all content in this study.

1.1 Background and Importance

In the last few decades, smartphone is an importance device in human's lives that provides various applications, including web applications, to support users in various aspects. For examples, people can work on smartphone anywhere-anytime; people can play social media using their smartphones; people can occupy their financial services whenever they need through the e-service; people can entertain themselves via entertainment software or webs. As the Internet growth, the security on web applications or websites via mobile phone becomes critical because there is many automatic malicious software attacks those applications. This malware normally emulates to be a formal user who comes for daily login. After passing through the login process, it can perform unpredictable actions to the system that may cause a large damage to the organization. That is the reason for those websites in using CAPTCHAs to prevent that malicious software. CAPTCHA is a test that is designed as a simple problem, which can be solved quickly by human, except malicious software. In the early 2000 [1], the CAPTCHA was displayed in a simple text form, presented with a set of characters that is embedded with noises, distortion, and complex background. Users took a few times, 10-15 second, to solve a CAPTCHA [2]. Many researchers proposed a variety of CAPTCHAs to break malware, such as Text-based

CAPTCHA, Image-Based CAPTCHA, Audio-based CAPTCHA, Game-based CAPTCHA, or Math-based CAPTCHA.

This research focuses on Text-based CAPTCHA according to its widely used to protect malicious software. In addition, Text-based CAPTCHA is the simplest form among other CAPTCHA's technique, as well as it is easy to be implemented and not required much computational resources in the automatically generated CAPTCHA. So, this CAPTCHA security is a low-cost CAPTCHA [3]. Moreover, input the characters of the Text-based CAPTCHA is straightforward via keyboard as QWERTY keyboard.

However, Text-based CAPTCHA was designed to prevent automated registrations, the attackers, or all unauthorized users, attempt to gain access into the website with the powerful of technology. There are two major approaches those are implemented to solve Text-based CAPTCHA. The first one, using OCR (Optical Character Recognition) technology to defeat Text-based CAPTCHA [4]. The OCR program recognizes the CAPTCHA characters and identifies the characters into the answer region afterwards. Commonly Text-based CAPTCHA is often based on English letters and numeric, so the performance of OCR technology can recognize those complicated. The second, the third-party CAPTCHA solving services, this is a business that hires human to solve CAPTCHAs with low-cost payment [5]. This approach recognizes characters of CAPTCHA by employing a group of people to read the CAPTCHA and send back the results. Currently, there are several CAPTCHA solving services in the market, such as "2captcha"; (<https://2captcha.com>), "anti-captcha"; (<https://anti-captcha.com/mainpage>), or "deathbycaptcha"; (<https://www.deathbycaptcha.com/user/login>). Consequently, the benefits of using CAPTCHA are reducing. Thus, merging some features, such as biometrics, to CAPTCHA may increase the ability to differentiate human attackers from the real human users.

Biometrics is an efficient technique that provides a superior security to against human attackers and other tricky-smart systems. As researchers signify variety differences biometric information from human characteristics on human bodies, it can be used as metrics for authentication. Biometrics can be classified into two groups: physiological biometrics and behavioral biometrics. The physiological biometrics related to the human body; the commonly known items are such as face, fingerprint, hand geometric, retina and iris. The behavioral biometrics referred to behavior of individuals, for example speech pattern, signature, or keystroke dynamics. As a result, the biometrics authentication method uses a part of human's body to identify a person with high accuracy to detect the person.

Over the past decades, desktop, PCs, or laptops were mostly used to access websites over the Internet. Thus, the technique of keystroke dynamic had been proposed to be used in the authentication process. Keystroke dynamic referred to the typing performance over the computer keyboards; this value is individual, and unique for each typing style, so it is difficult to be duplicated. There are various variables that can be measured in different aspects, such as typing time duration, typing speed, or percentage error on typing pattern; these values can be used to identify users. Furthermore, the most advantage of keystroke dynamic is that it is inexpensive and does not require any addition devices.

So, keystroke dynamics refers to the automated method for identifying or confirming the identification of a person based on the user's individual typing rhythm on the keyboard. Keystroke dynamics analyzes the typing behavior over characters that each user types and distinguishes users using keystroke's factors, such as overall speed typing, duration between key press or key release, pressure on the screen when

holding the key, and size of the finger, etc. So, this technique is impractical to be copied by observing. Moreover, keystroke dynamics is easy to be obtained by backend software implementation as a low-cost security solution.

Recently, smartphones and tablet devices with touch screen features were popular and the most frequency used. In addition, the touch screen also has functions to generate a traditional keyboard on the screen for users who are familiar with it. Therefore, special factors can be measured as additional variables to identify individual users; these are fingertip pressure, fingertip size, and position pressure key.

According to the serious human-based attack on CAPTCHA security, this paper presents a new approach of Text-based CAPTCHA that is derived from individual keystroke dynamics to classify a legitimate users or attackers. The aims of this paper are providing the secure of Text-based CAPTCHA to protect websites from human attackers, and identifying legitimate users using special derived keystroke dynamics. Moreover, this paper investigates possibilities to identify person by considering from the individual typing on the virtual keyboard.

1.2 Objective

- To determine personalize features for developing an individual Text-based CAPTCHA over the touchscreen device.
- To develop a mechanism that automatically generates an individual Text-based CAPTCHA for an individual user.
- To determine the most suitable validation model for the security validation process.

1.3 Scope of the research

This research selectively focuses on the study on the effective identification of a combination of factors obtained from the smartphones. The experiment was designed for smartphones with iOS operating system and distributed via the web application.

Furthermore, the experiment is designed based on iOS that can produce finger touching area value at each touch. However, this design can be applied on any smartphones because the used functions are independent from the experimental smartphone.

1.4 The expected outcomes

To obtain a mechanism that automatically generates a Text-based CAPTCHA which relies on independent users over a touch device.

1.5 Definition

Smartphone:	Touchscreen mobile phone device
Keystroke feature:	The feature is extracted from pressing the character of the bottom on the screen.
Key down:	The event of time that user pressing the character of the bottom on the screen.
Key up:	The event of time that user releasing the character of the bottom on the screen.
Dwell time:	The duration time of the user pressing the character of the bottom on the screen.
Interval time:	The duration time between the release the current bottom and the next bottom.
Latency time:	The time interval release of the bottom and press the next bottom.
Flight time:	The time interval between press of the current bottom and press the next bottom.
Up to up time:	The time interval between release the current bottom and press the next bottom.
Fingertip:	The size of finger while user press the character of the bottom on the screen.
Pressure (Radius):	The force of finger while user press the character of the bottom on the screen.
Location of typing:	The location of finger while user press the character of the bottom on the screen.

Offset x:	The <i>x coordinates</i> of the user <i>finger's</i> pressing with the visual keyboard.
Offset y:	The <i>y coordinates</i> of the user <i>finger's</i> pressing with the visual keyboard.
Page x:	The <i>x coordinates</i> of the user <i>finger's</i> pressing with the <i>screen</i> .
Page y:	The <i>y coordinates</i> of the user <i>finger's</i> pressing with the <i>screen</i> .
Accuracy [6]:	The number of correct predictions over the output size.
Recall [6]:	The quantifies the number of positive class predictions made from all positive examples in the dataset.
Precision [6]:	The quantifies the number of positive class predictions that belong to the positive class.
WPM (Word per minute):	A measurement of the speed of typing
Time:	Time measurement in millisecond
User:	The person who uses the touchscreen devices

1.6 The structure of thesis

The remaining chapters consists of the related works that briefly describes the previous studied in Chapter 2. The details of the research methodology are elaborated in Chapter 3 while the results of the research are displayed in Chapter 4,

following with Chapter 5, 6, 7 as the discussion, conclusion, and the future work, respectively.



Chapter 2

Literature Review

2.1 CAPTCHAs

The power of World Wide Web changes human's lives because, currently, people mostly rely on online service websites. Thus, websites are unauthorized access under various reasons, especially bots' attack. So, in 2004 [1], Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA) was introduced as a safety mechanism, that was used to protect unauthorized access under various reasons, especially unwanted program. The idea of CAPTCHA comes from the Turning test, that is the challenge testing by asking users to solve simple tasks before accessing to a website, these tasks can classify human from bots. [4, 7]. So, CAPTCHA is a program to protect websites from malicious software by generating and grading test that only human can pass. The technique, firstly, presents series of letters which can be either alphabets or numeric generated by an automatic mechanism with noise and distortion. Hence, this designed CAPTCHA can distinguish between the Internet users and bots because it is too difficult for bots to solve [8].

At present, CAPTCHA is mainly used in the domain of network security to determine users and unknown, based on questions and answers. So, variety question's types of CAPTCHAs have been proposed and implemented, such as pictures selecting, text entering, or garbled sound solving. Generally, there are 3 main CAPTCHA questions: Text-based CAPTCHA, Image-based CAPTCHA, Audio-based CAPTCHA. Besides, these CAPTCHAs, there are some other types of CAPTCHAs, such as Game-based CAPTCHA, Math-based CAPTCHA, etc. Though there are various types of CAPTCHAs have been proposed, Text-based CAPTCHA is the most widely used because it is friendly with

users, and easy to be implemented. Amongst the different types of CAPTCHAs, this research focusses on the simplest and the most used as Text-based CAPTCHA.

Text-based CAPTCHA is the most common and the simplest used in several websites. Sadly, every day, more than a hundred million of Text-based CAPTCHAs are solved all over the World [9]. The method of this type of CAPTCHA starts with randomly generated letters; and users must enter the letters like that generated CAPTCHA. Moreover, Text-based CAPTCHA can be much complicated for bots by applying various font sizes, lower and upper case, adding noises, distortion, and using complex background [10].

On the other hand, image-based CAPTCHA is also used in the various websites because bots cannot solve the vision. This image-based method requires users to perform actions such as select the right images with the label, identify pictures, matching images, or even, rotate the image to the right orientation [11-13]. The advantage of this kind over the text-based CAPTCHA is just a click to confirm that users are not a bot. Nevertheless, the image-based method is not suitable for users who have visual impaired and blindness. Consequently, the implementation of the audio-based CAPTCHA had been proposed.

As the fact that the audio-based CAPTCHA had been implemented as an alternative [14] and also to support human visual impaired and blindness user [15]. This CAPTCHA requires users to listen to a clip of a letter's stream; this clip includes noise, distortion, and interference. Moreover, only English language is used for this technique. Thus, English must be either users' mother tongue or their second language. Another weakness of this method is the hearing difficulty of some users. Nonetheless, the vocalization quality of this CAPTCHA is low until the users cannot

understand easily. In addition, this system uses only English language which might not be familiar for users in non-English speakers.

Moreover, out of these three main methods, there are some others such as such as Video-based CAPTCHA, Puzzle-based CAPTCHA, Game-based CAPTCHA, or Mathematic-based CAPTCHA are proposed to be alternative to users [16, 17]. Some researchers, as [18] and [19] had proposed some Game-based CAPTCHAs that challenge users to response the test based on the granted Game. So, this technique provides a fun task for users.

2.2 Overview of Breaking

2.2.1 Breaking Traditional Attackers

Currently, the authentication is vital because it protects unauthorized persons in entering to significant area or gaining access to priceless devices and information. In the early age of Internet and computing, the common way to identify a person is to use the personal identification number, as known as PIN; this method was introduced by John Shepherd-Barran, a managing director of London's De La Rue Instruments in the year 1964 [20]. Currently, PIN is commonly used in the banking system with the length of 6 or longer, smartphone, and more; some PIN will be encrypted while used to prevent hackers. Although PIN is flexible as a multipurpose method, it is easy to be hacked because it is easy to be guessed [21, 22] or easy to be surfing attack [23, 24]. Another trick of the hacker is to put an extra camera to record the victim's PIN at the ATM [25], including add a fake pad on top of the key pad of the machine or ATM "wiretapping" or "eavesdropping" [26, 27]. Password is another alternative in the authentication process that has been used in the login process for decades. One defect of using password is that it can be copied easily by the intruder because of

improperly setting. From the reports of [28] and [29], the top five favor passwords are “123456”, “123456789”, “qwerty”, “password” and “111111”. Moreover, [30] and [31] had confirmed that the password from user’s birthday or family name can also be easily guessed while [32] and [33] had reported that behaviors like writing down the password, the reused password, and the long-term password also cause unsecure for the password. Later, the pattern authentication had been introduced according to the development of touchscreen devices. Regrettably, this method is broken by guessing attack, shoulder surfing attack, smudge attack, and side channel attack [34]. The high rate achieves of guessing the pattern login is up to 74.17% when combining with the smudge attack [35]. However, using of many line-visibilitys and pattern lengths can protect the mobile device accessibility from malicious hackers [36].

According to the weakness of fundamental authentication techniques, the use of biometrics had been introduced around 1890s [37]. The biometrics can be classified to two types: physiological and behavioral; each of them was chosen for security mechanism. Some common the physiological characteristics are iris, fingerprint, face, and retina while the well-known behavioral characteristics are keystroke, typing speed, and voice. Presently, single biometrics is often used in various types of components, such as computer, tablet, smartphone, room, etc. A well-known fingerprint authentication method has been used for many years in many computer devices, especially notebooks. This method uses the truth that people will not have the same fingerprints although they are twins; the ridge on each finger is permanent and the tip pattern is absolutely unique [38]. Thus, the performance of matching fingerprints has accuracy more than 99%, and the false positive rate of 0.1% [39, 40]. Therefore, in 2013, high standard smartphones as Apple, Samsung, HTC, and Huawei have integrated fingerprint sensor as a part of their products [41]. Moreover, fingerprint has been implemented to enhance the payment process [42, 43]. Apart

from the daily life usages, the fingerprint has been used as an evidence in the criminal justice system [44], tools for managing classes enrollments and attendances [45], and an electronic devices identification [46]. However, some personal diseases have effects to the fingerprint usages, such as a very soft fingerprint, or palm's peel and the ridges are fading out or altering from time to time. Moreover, other poor entering fingerprints conditions are dry, wet, oily, dirty fingers, and scarred [47]. So, fingerprints applications are related to law, immigration, and banking or financing [48]. Though using fingerprint has many benefits and high secure because it is hard to emulate. In some cases, the fingerprint authentication may be failed to detect the identity of the person because the sensor fails to capture the fingerprint; 20-25% of people have this problem [49]. Moreover, from the report of [50] and [51, 52], the ability of the malicious application to imitate the owner's fingerprint is closed to 80%. Besides, in the case of the decoding attack, the hackers remoted to steal fingerprints from the target's smartphone, sent the template of fingerprint from service stores embedded in the smartphone via email, then, decoded the template of fingerprint for fully reentering the system as the owner [53].

As people say, "Eyes are door to everything.". This statement may be right when the system uses iris and retina as the secret key to enter a system or to gain access to a device. The use of either one of these two components is called as the eye authentication. From the patterns of retina, these patterns are unchanged for the entire life of the person [54-56]. So, the level of accuracy based on the retina identification is 97.50% [56]. Like the retina, iris is a color-thin tissue inside the human's eyes that is composed from many layers of color-tissues that form as smooth muscle around the pupil where the unique pattern of each iris can be created and stay through the lifetime. Thus, the result from the research of National Institute of Standards & Technology (NIST) indicated that the accuracy of iris identification is 90-99% [39]. Consequently, both retina and iris have been used for

high security organizations, such as several high levels of government in U.S., military, FBI, CIA, and NASA [57, 58]. Nevertheless, there are cases of eye illnesses or eye diseases that can degrade the characteristics of retina; so, the eye detection fails.

From the contents of biometrics mentioned above, it is clearly seen that using biometrics do not completely free from a detection error because of unavoidable problems, such as malicious software, and deceases. Therefore, using only single biometrics for security control has been re-considered while the multifactor authentication has been proposed. Nevertheless, using various factors from combination of biometrics still has no guarantee that there is no chance to be intercepted by the imposters. Thus, this research proposed a new method to protect personal information based on combination between personal profile and biometrics factors. The expected outcome is a new protection model based on the user-profile that cannot be copied by any others; in addition, these proposed factors will not be altered even though the user has decease, or the system is attacked by malicious software.

2.2.2 Overview of Breaking Methods towards Text-based CAPTCHA

Although several CAPTCHA techniques have been proposed to protect system from bots, most users always have experiences with the Text-based CAPTCHA more than others. Even so, most complex Text-based CAPTCHA can be extracted by some malicious software because of the evaluation of Information Technology. So, this section describes the breaking techniques for the Text-based CAPTCHA that have been used in the certain period.

1) Random guessing attack

Random guessing attack is as known as the Brute Force attack that is a common threat of cybercrimes. A brute force attack attempts to guess the possible letters, numbers, or extra symbols of the Text-based CAPTCHA questions until discovering the correct word, then they will gain access to the system. The 100% of success rate with automatic guessing attack, for example, 4-digits numeric of Text-based CAPTCHA would have 1,000 possible answers, if there is no time limited efforts to solve [59].

2) OCR recognition

OCR recognition stands for Optical Character Recognition software is used to read text that was presented in Text-based CAPTCHA question [60]. The general of OCR technique starts from extracting CAPTCHA's characters, matching each extracted character's image with the image of character stored in the Character-Corpus. Clearly, using OCR recognition usually provides high success rate in breaking the Text-based CAPTCHA [61]. The findings indicated that nearly 100% success to break Text-based CAPTCHA relies on a high-quality OCR program [61, 62].

3) CAPTCHA farm

CAPTCHA farm is the CAPTCHA solving business that has been settled for a decade since Google and YouTube requested to prove the humanity of the users. This is an unlawful business that usually employs human in a cheap payment to do the work. Using API captured the image of CAPTCHA from the screen, sent it to a group of human solvers, when the CAPTCHA was solved by the solvers, then it was sent back to the bot to fill in the test. So, this business claims that 100% of CAPTCHAs have been solved by real human from around the world, such as Indian, Pakistanis, or Vietnamese, with very low price; 1 USD per a thousand words of CAPTCHAs [63]. Consequently, the CAPTCHA farm is a severe human attack issue that many researchers are concerned because CAPTCHAs were broken by the 3rd party human instead of bots [5, 64-66]. As a result, the system cannot be protected by CAPTCHAs.

According to the emergence of human CAPTCHA solving business, many new types of CAPTCHAs were introduced as CAPTCHA challenges to protect this illegal business. For example, iCAPTCHA or Interactive CAPTCHA [67] was presented to protect the human CAPTCHA solving. This system uses a simple Text-based CAPTCHA that is human's friendly. In this case, the statistical timeline is applied to justify the user and the human attackers. Since the real user will see the CAPTCHA, the solving time will be in the acceptance boundary of the system; on the other hand, the scripts of the intruders from CAPTCHA farm spent time in capturing the CAPTCHA's image, sent to the group of the farm to solve, waited and received the answer, total process time of this bot is definitely higher than the authenticated person. Another idea had been proposed, the GeoCAPTCHA, this aims to protect the 3rd party human attack [68]. This technique uses the rotation concept to obtain the right answer as the users must rotate 3D street-view image of Google map to be the same as 3D street-view image of Google map in the register step. So, the rotation of attackers cannot be identical to the real one. Later, in 2013, Ye, et al. [69] had designed DDIM-CAPTCHA

or Drag-n-Drop Interactive Masking CAPTCHA to break traditional attack and human attack. Furthermore, it is also able to protect the OCR technology recognition. This technique uses benefits of the tri-dimension space to contain the variety of letters that can be overlapping; the users just drag letters from different layers according to the test question, and then, drop the letters to the answer's region. Another interesting game is the Stackelberg security game that uses a simple test to distinguish the human and bots. Additionally, this technique also uses the response time value to determine the authenticated user and attackers from the CAPTCHA farm because the response time from the authenticated user will be definitely less than the response time of the CAPTCHA farm [70].

Besides the response time that is used to distinguish between human and bots, there is another distinguishing technique that had applied physiological biometrics with the CAPTCHA, called as HandCAPTCHA. This HandCAPTCHA is an approach for hand images that used the physiological biometric recognition based on hand images that was implemented over the traditional Text-based CAPTCHA. The ability of this method is it works against malicious threats with 98.34% accuracy [71]. In 2020, the keystroke features integrated with the IP address of the user are applied to classify a person under the use of Text-based CAPTCHA. This method can detect the human attack because the IP address of the user's device can scope down the imposters [72].

Table 1 presents a brief description of previous study mentioned. Most of the past works that study CAPTCHA to against automated bots and the 3rd party human attack.

Table 1 List of CAPTCHAs that can protect the 3rd party attack

CAPTCHA	Type of CAPTCHA	Technique	Performance
iCAPTCHA [67]	Text-based CAPTCHA	Using statistic timing of solving CAPTCHA	0.00% of false negative error rate, and 1.77% of false positive error rate
GeoCAPTCHA [68]	Image-based CAPTCHA	Rotated 3D street view	Testing with automatics attack is fail, testing with other human solving is fail, and testing with user is pass
DDIM-CAPTCHA [69]	Text-based CAPTCHA	Drag the letters from 3 layer and then drop in the answer box	99.00% accuracy.
Stackelberg game [70]	Game-based CAPTCHA	Using statistics of latency time	the break-even points of whether adopting machine solver or human solver can be determined.
HandCAPTCHA [71]	Image-based CAPTCHA	Geometric	99.5% accuracy, 3.39% of EER.
A new methodology for detecting human attacks on text-based CAPTCHAs [72]	Text-based CAPTCHA	Based on the Euclidean distance of Keystroke dynamics	-

2.3 Keystroke Dynamics

Keystroke Dynamic (KD) is a behavior biometrics that uses user's timing information during typing a computer keyboard. Currently, many researchers use KD systems in the enrollments and the verification step via using input timing information from pressing each key of the computer keyboard. Timing features that use in KD, such as Di-graph latency, down-down, up-down, hold, Tri-graph latency, and N-graph latency [73]. So, time usage was studying in numerous investigations. Monroe and Robin [74] used keystroke features of typists, such as Di-graph latency and time duration, to identify the users.

As statistics approaches are such as this system. Besides, Hempstalk et al. [75] used keystroke features such as di-graph latency, key duration, typing speed, and error rate of user typing. This technique estimated the performance of combination approach technique as density estimation and a class probability. Another researches based on neural network, Sungzoon Chu et al. [76] proposed using individual password and time recording to identify verification. Moreover, Villani [77] analyzed a long text input via using Euclidian distance and k-nearest neighbor. This research used a java applet to record keystroke data such as di-graph latency, key duration, typing speed, a percentage of special characters and editing pattern, then the result indicated that 99.5% success on 36 samples.

So, keystroke dynamics is a biometrics feature based on typing behavior that can be used to identify individuals. So, it focuses on the typing patterns of users on the QWERTY keyboard. Since the year 2002 [78], the original measurement of keystroke dynamics has found 4% of false acceptance rate and less than 0.01% of imposters gain access to the system via their own dataset. From the study in 2004, the performance of keystroke dynamics is confirmed as 0% false accept rate and less

than 1% of valid imposters as a real user [79]. In addition, when the keystroke dynamic was applied with the finger pressure for authenticate a person over a touch pad of a notebook by entering 10 digits phone number, the accuracy is as high as 99% [80]. This study indicates that the combination of keystroke dynamic and other biometrics can increase the detection accuracy of the authentication system.

Thus, for smartphone, this keystroke dynamic can be used to classify a person as same as keystroke dynamics on the computer keyboard. Besides, this technique can be use with the sensor on mobile devices to detect the typing rhythm, so the detection equipment is cheaper than another biometrics authentication. Table 2 presents the performance of using keystroke dynamic on touchscreen devices.

Table 2 The related studies on keystroke dynamics based on touchscreen device

Researcher	Feature	Number of participants	Entering	Performance
[81]	Keystroke feature, pressure, finger size	152	17digits passphrases	4.59% FRR, 4.19% FAR
[82]	Hold-key, pressure, touch area, location of keypress, device orientation	13	passphrases	14% FRR, 2.2% FAR
[83]	Screen location, pressure, key-press time and key-release time, gyroscope, and accelerometer	52	10 digits	3.9% EER

Another research that combined keystroke features, finger pressure, and finger size, and working on a touchscreen smartphone using 152 samples, showed that only 4.59% of the false reject real users, and 4.19% of the false accept imposters [81]. Moreover, another study had collected the mobile behavior of user's key press information as hold-key, pressure, touch area, location of keypress, and device orientation. This study worked on 13 samples and let them use a soft keyboard on a mobile device. The results found that the detection rate of the system was 86%; moreover, false accept rate and false reject rate were 14% and 2.2%, respectively [82]. According to the studied on 52 samples of [83], the values of time of key-press and key-release, pressure, screen location, gyroscope, and accelerometer were recorded and used as indicator to classify a person from human attack. When only the times of keypress and key-release were used as two the authentication key factors, the equal error rate (EER) was 19.7%; when adding the touchscreen features as pressure, screen location, the EER was dropped to 4.0%. In addition, adding all features to classify a person from the intruder, the result gained 3.9% for EER.

Chapter 3

Methodology

This chapter comprises with 3 main sections. The first section explains the feature analysis, and the second describes while the last section is the proposed of a new Text-based CAPTCHA.

3.1 Preliminary Test

3.1.1 Program for finding differences between right-handed and left-handed

This research had developed a program to collect data for finding significant factors between persons who are right-handed and left-handed. The program was written using JavaScript to store keystroke time by let the samples type random characters shown on the screen, as shown in Figure 1.

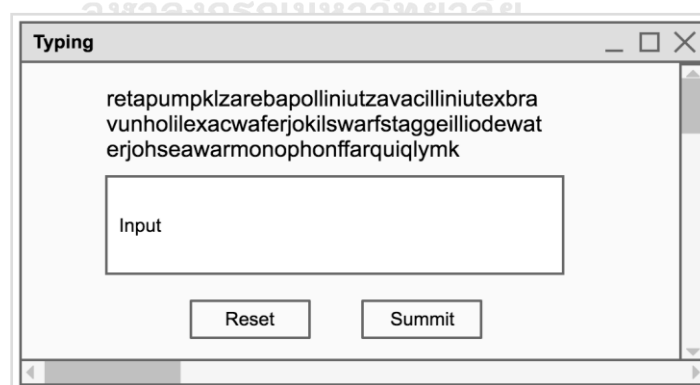


Figure 1 Collecting screen and text

The test was performed using 12 randomly selected samples: 6 left-handed and 6 right-handed. Collected data had been analyzed using 2 independent sample *t*-test

to find the typing rhythm over the computer keyboard. The expected was the differences between the typing rhythms of these two groups. The analyzing result can conclude that the typing rhythm of both groups had no significant differences. All results are shown in Table 3.

Table 3 Results of Independent Sample T-test

ID	Sig.				
	Dwell	Interval	Latency	Flight	Up to up
1	0.170	0.165	0.119	0.119	0.125
2	0.580	0.001	0.001	0.001	0.001
3	0.748	0.331	0.909	0.724	0.684
4	0.000	0.001	0.001	0.001	0.001
5	0.001	0.433	0.304	0.343	0.374
6	0.395	0.000	0.000	0.000	0.000
7	0.578	0.385	0.398	0.414	0.390
8	0.148	0.537	0.324	0.384	0.455
9	0.374	0.555	0.612	0.501	0.623
10	0.512	0.640	0.537	0.559	0.626
11	0.902	0.578	0.643	0.630	0.579
12	0.602	0.211	0.212	0.199	0.235

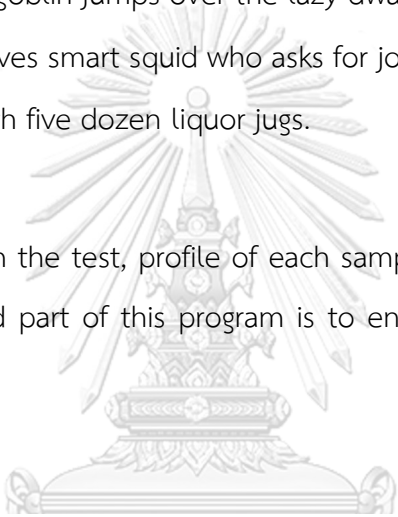
3.1.2 Program for collecting the typing rhythm

The developed program will display all 26 English characters in one sentence. Each sample must type 5 sentences without replicated. The sentences are listed as follow.

- The quick brown fox jumps over the lazy dog.
- Jackdaws love my big sphinx of quartz.
- The quick onyx goblin jumps over the lazy dwarf.
- Cozy lummoX gives smart squid who asks for job pen.
- Pack my box with five dozen liquor jugs.

Before samples perform the test, profile of each sample must be entered as shown in Figure 2. The second part of this program is to enter the sentence, as shown in Figure 3.

User Information



Name	<input type="text" value="Name"/>
Email	<input type="text" value="Email"/>
Telephone	<input type="text" value="Telephone"/>
Are you	<input checked="" type="radio"/> Male <input type="radio"/> Female
You are	<input checked="" type="radio"/> 10-18 <input type="radio"/> 19-24 <input type="radio"/> 25-40 <input type="radio"/> 41-65
Nationality	<input checked="" type="radio"/> Thai <input type="radio"/> Bhutanese <input type="radio"/> Indian <input type="radio"/> Chinese
	<input type="button" value="Reset"/> <input type="button" value="Submit"/>

Figure 2 The sample of the Profile page

Typing Test

The quick brown fox jumps over the lazy dog.

Figure 3 Sentence Entering page

From the fundamental analysis by comparing the typing time of each character using Dwell time to find the differences among samples, the result as shows as in figure 4 that the means of Dwell time of each character of each sample is significantly different. Therefore, it can conclude that each person has different typing rhythm.

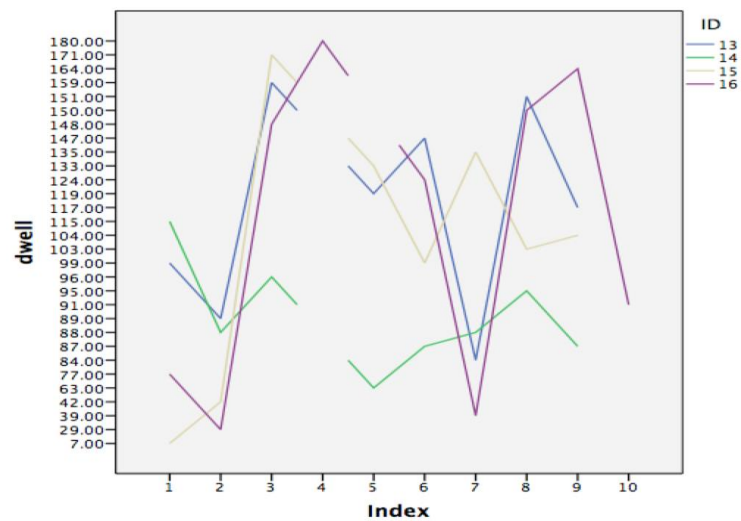


Figure 4 Differentiation line graph of all samples

3.2 Feature Analysis

This section describes the methodology that has been implemented in this research to obtain required factors for a profile-based security mechanism. Therefore, details in this section will be divided to 4 procedures as the researcher has executed in this experiment. The following diagram in Figure 5 shows all 4 executed procedures; and details of each procedure is elaborated in the following subsections. This section includes the results of feature analysis.



Figure 5 Steps of all 5 executing procedures in this experiment.

3.2.1 Data Collection

The data collection is the first process to obtain data for creating the classification model for a profile-based authentication mechanism. In this experiment, the data collection method is volunteer sampling where the samples this experiment are 3rd year undergraduate students from Computer Science program, Faculty of Science, Chulalongkorn University. Thus, these students are familiar with high technology and all use high-tech smartphones. This research intends to propose strong and undegradable factors for creating an authentication classification model so the system will be highly protection. From this aim, the factors of the model must be chosen from things that belong to the user for entire life and must be things that user is much familiar than others, such as full-name, surname, email, phone number, and gender. Thus, each sample was asked to enter gender, age-range, education-level, posture, name, surname, e-mail address, mobile phone, and gender, for 10

times repeatedly. The collected period was 10 working days using an application that has been written in PHP and JavaScript. The application for data collection is a web-based application that can run on both Android and iOS without any codes altering. The collected data will be sent immediately and automatically to a cloud system, no data will be stored in the testing smartphone. The data collection process is shown in Figure 6, In addition, the entering keyboard is the self-implemented keyboard that emulates the touchscreen keyboard.

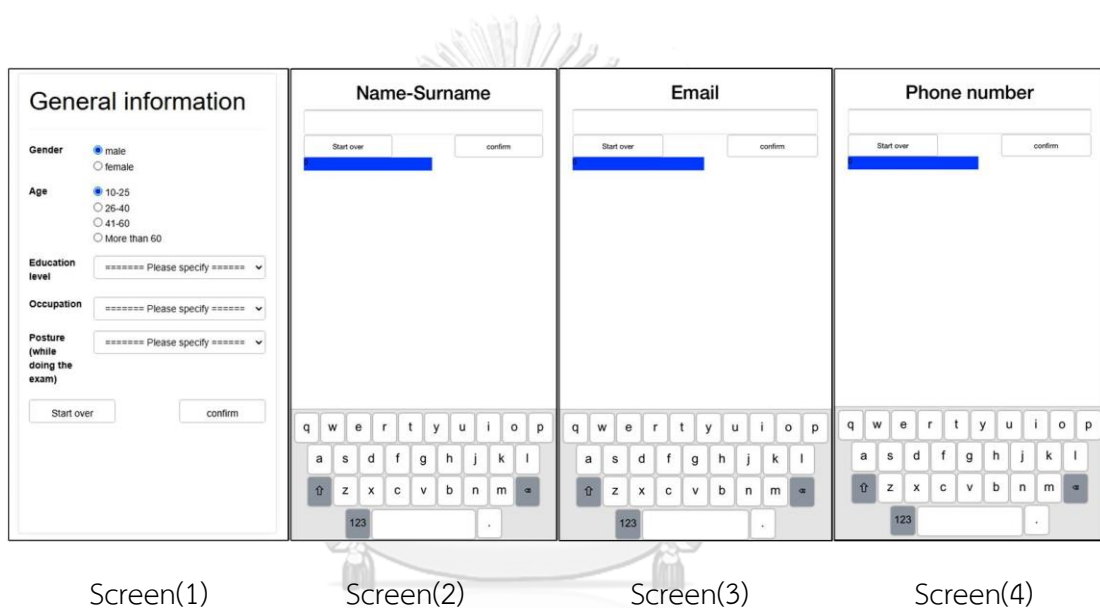


Figure 6 Data collection process.

Since the entering data are stored in the cloud system, the used database is PhpMyAdmin with 4 separated tables: user_information, full_name, email, and mobile phone number. In addition, the objective of stored data in each table is different; for example, the user_information aims to store general information of each sample, such as email, name, surname, phone, and gender, etc. For tables of full_name, email, and phone, these three tables have the same attributes and they are responsible for storing one character per record, the character is obtained from name and surname, email, and mobile phone number. Figure 7 demonstrates the Database diagram among 4 tables.

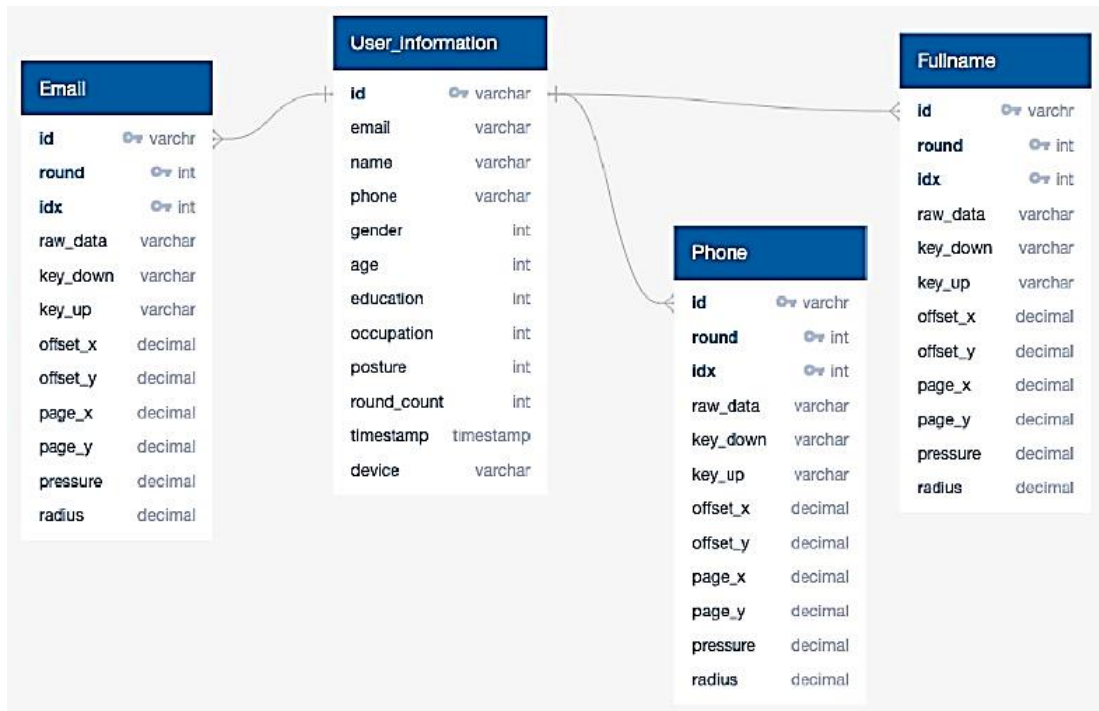


Figure 7 The Database Diagram on the Cloud

3.2.2 Preprocessing

After the data was collected and record to the cloud, these data were checked for suitability and completion for the analysis process. From the volunteer sampling in 10 working days, there were 40 volunteers: 32 males and 16 females. Therefore, all data of 40 samples in the cloud were exported to the .xlsx format. Then, outliers and missing values were checked and eliminated from the entire data set. The criteria to eliminate the outliers is based on 95% confident level where the Empirical rule is applied as 95% of all tested elements will fall in the area between 2 standard deviations as shown in Equation 1, Figure 8 shows the acceptant interval of the valid data set in this experiment.

$$(\bar{x} \pm 2sd)$$

Equation 1

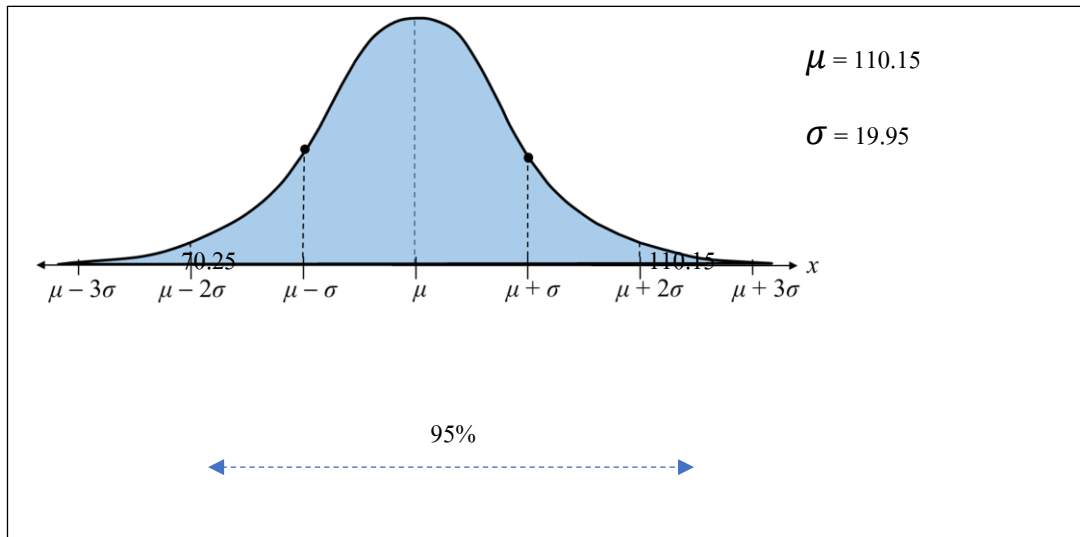


Figure 8 The acceptant interval of the valid data set

Once the elimination of the outliers from the entire dataset, the next step is to delete all incomplete records. The conditions for deleting the incomplete record can be classified in 2 cases. The first case is the entering data was missing according to some communication errors, or the second case is the sample was interrupted and reentering data from the first step again. Thus, according to both conditions, the previous data was not complete and must be deleted from the data set.

CHULALONGKORN UNIVERSITY

The next step is to delete the special character, including space between words. The special characters are such as at sign (@), comma (,), and full stop (.). These characters can usually be found from email of samples. For example, bpattara@yahoo.com must be eliminated all special characters to be bpattarayahoocom and stored back to a local database.

Another step that must be executed is to count the frequency of each character entering from each sample. If any characters have frequency within the rank of 10, this character will be used in the feature extraction process; otherwise, it will be ignored.

In addition, all keystroke features, such as dwell time and interval time, were calculated in this preprocess. Figure 9 shows the computation of all keystroke features. Once all computing information were calculated, the gender of the sample is encoded as 1 for male and 2 for female.

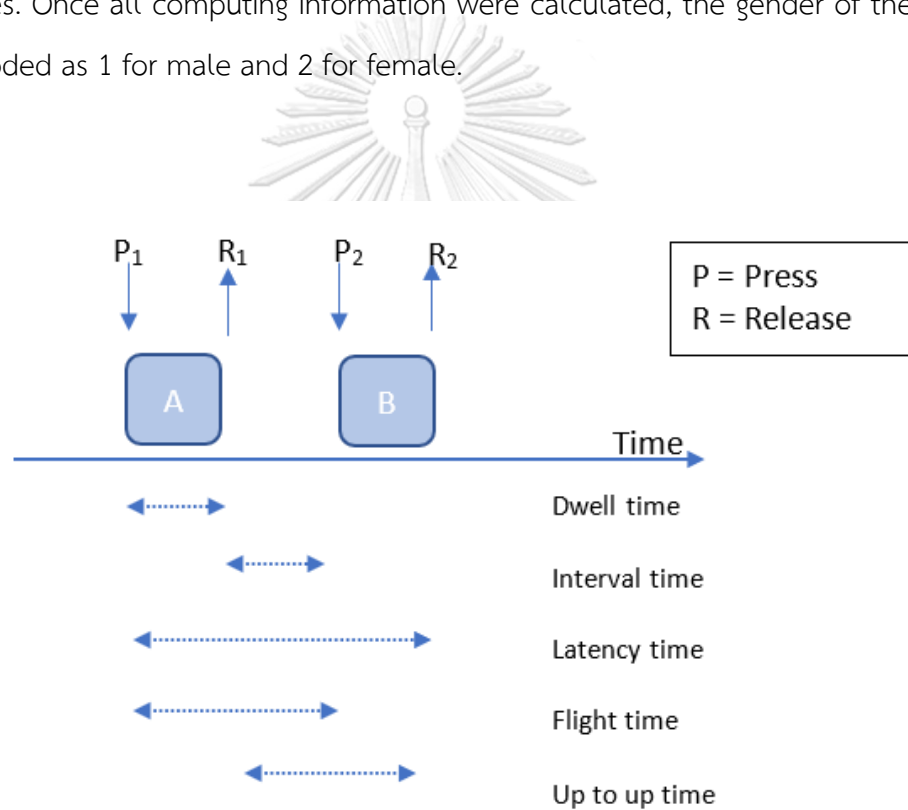


Figure 9 Demonstrations of the keystroke features

The results from the pre-process are 4 separated tables in PhpMyAdmin at Cloud system. However, the size of each table is not equal because of its spelling. These files will be used as input of the next step, feature extraction.

3.2.3 Feature Extraction

This process is an important part because all significant features related to the model creation are established. In this process, two methods are deployed to break out the vital features from the entire attributes in the cloud database. Since the distributions of all data set are not normal, therefore, the nonparametric test is applied to filter out the suitable features using 95% confident level. In addition, in the statistical process, the data set to be tested is the set of single character with the keystroke features related to the character, including gender of the sample. The focusing elements are the characters that have frequency in the first 10th rank of the entire typing.

To perform the statistical testing, the confident level for all testes in this experiment is 95%, or 0.05 significant level (α). Moreover, the normality test must be executed to confirm the normal distribution of all factors before working on relations among factors. Firstly, from the researcher's observation, there are two main factors that may have impact to other features, especially, the keystroke features' values. These two factors are the fingertip area, and gender of the sample. Therefore, testing of effects from both factors are performed based on the characteristics of their data.

1) *Test for Gender's effect*

According to the observation, Gender has potential to be a factor that can affect to the keystroke features which can combine with gender to create the unique personality of a person. So, to prove such assumption, 2 independent samples test, such as *t*-test or Mann-Whitney U test, is deployed using 95% confident level. All hypothesizes are listed below.

The first test is to confirm the believe that the duration of pressing a button of male and female does not have the same value. Thus, the hypothesis can be setup as follow.

$H_{01}(\text{Dwell} \times \text{Gender})$: There is no significant difference of mean values of the dwell time between male and female.

$H_{11}(\text{Dwell} \times \text{Gender})$: There is significant difference of mean values of the dwell time between male and female.

The second test is to confirm that male and female move their fingers in different speed while typing two characters.

$H_{01}(\text{Interval} \times \text{Gender})$: There is no significant difference of mean values of the interval time between male and female.

$H_{11}(\text{Interval} \times \text{Gender})$: There is significant difference of mean values of the interval time between male and female.

In 2015, Jason Lui had reported that the typing speed of male is faster than female when typed with computer keyboard. Therefore, with the touchscreen, this assumption should be tested whether male still types faster than female or not. Thus, three null hypotheses below have been drawn as follow.

The third assumption is to test for the believe that the latency value of male and female may be dissimilar.

$H_{01}(\text{Latency} \times \text{Gender})$: There is no significant difference of mean values of the latency time between male and female.

$H_{11}(\text{Latency} \times \text{Gender})$: There is significant difference of mean values of the latency time between male and female.

The fourth assumption is to test that the flight time value of male and female may be dissimilar.

$H_{01}(\text{Flight} \times \text{Gender})$: There is no significant difference of mean values of the flight time between male and female.

$H_{11}(\text{Flight} \times \text{Gender})$: There is significant difference of mean values of the flight time between male and female.

The fifth assumption is to test that the up-to-up time value of male and female may be dissimilar.

$H_{01}(\text{Up-to-up} \times \text{Gender})$: There is no significant difference of mean values of the up-to-up time between male and female.

$H_{11}(\text{Up-to-up} \times \text{Gender})$: There is significant difference of mean values of the up-to-up time between male and female.

The sixth assumption is to test that the finger's strength of male and female may not be equal. Moreover, the finger's strength of male should be stronger than female so the value of pressure of male should be greater than female. Thus, the hypothesis to test such suspicious is follow.

$H_{01}(\text{Pressure} \times \text{Gender})$: There is no significant difference of mean values of the pressure between male and female.

$H_{11}(\text{Pressure} \times \text{Gender})$: There is significant difference of mean values of the pressure between male and female.

Once the finger's strength of both genders is suspicious for not equal, then the seventh assumption is to test that the fingertip area of male and female may not be equal according to the different strength of the fingers.

$H_{01}(\text{Fingertip} \times \text{Gender})$: There is no significant difference of mean values of the fingertip area between male and female.

$H_{11}(\text{Fingertip} \times \text{Gender})$: There is significant difference of mean values of the fingertip area between male and female.

2) Test for Fingertip's effect

The objective of the test is to prove that if the testing factor can influence the change of the keystroke features' values, then it can conclude that the keystroke features and the tested factor has potential to be in the classification model. Therefore, the n -independent samples test, such as Complete Randomized Design (CRD) or Kruskal-Wallis test, is performed to test the effect of the fingertip towards each keystroke feature; those are finger pressure, dwell time, interval time, latency time, flight time, and up-to-up time. Consequently, there are 5 main hypotheses to be analyzed as written below.

The first hypothesis aims to measure the effects of the fingertip area towards the value of finger pressure. This means, if a person has large fingertip area, the person should have high value of the finger pressure. On the other hand, large fingertip area should be able to imply that the finger pressure should be high. In such case, the hypothesis can be setup to,

$H_{01(\text{finger pressure} \times \text{fingertip})}$: There is no significant different among means of the finger pressures when the fingertip areas are different.

$H_{11(\text{finger pressure} \times \text{fingertip})}$: There is at least one significant different of mean of the finger pressure when the fingertip areas are different.

The second hypothesis aims to measure the impacts of the fingertip area towards the dwell time. This means, if a person has a large fingertip area, there is a possibility that the dwell time should be longer than the person with a smaller fingertip area. In such case, the hypothesis can be setup to,

$H_{02(\text{dwell time} \times \text{fingertip})}$: There is no significant different among means of the dwell time when the fingertip areas are different.

$H_{12}(\text{dwell time} \times \text{fingertip})$: There is at least one significant different of mean of the dwell time when the fingertip areas are different.

The third hypothesis aims to prove that with the large fingertip area, the large value of the latency time can be obtained. So, the hypothesis can be setup as follow.

$H_{03}(\text{latency time} \times \text{fingertip})$: There is no significant different among means of the latency time when the fingertip areas are different.

$H_{13}(\text{latency time} \times \text{fingertip})$: There is at least one significant different of mean of the latency time when the fingertip areas are different.

The fourth hypothesis is to test that for different fingertip areas, the values of the flight time can be different.

$H_{04}(\text{flight time} \times \text{fingertip})$: There is no significant different among means of the flight time when the fingertip areas are different.

$H_{14}(\text{flight time} \times \text{fingertip})$: There is at least one significant different of mean of the flight time when the fingertip areas are different.

The last hypothesis for the effect of the fingertip area is related to the values of up-to-up time. The hypothesis is to prove that if the fingertip area is large, then, the up-to-up time should be large. In such case, the hypothesis can be written as follow.

$H_{05}(\text{up-to-up time} \times \text{fingertip})$: There is no significant different among means of the up-to-up time when the fingertip areas are different.

$H_{15}(\text{up-to-up time} \times \text{fingertip})$: There is at least one significant different of means of the up-to-up time when the fingertip areas are different.

After the test of effect from the gender and the fingertip towards the keystroke features, the factors that will be selected for the classification and evaluation process are all that the results from statistical test are rejected. For example, if the result of the hypothesis “There is no significant different among means of the flight time when the fingertip areas are different.” has been rejected, so, it means, there is at least one significant different of mean of the flight time when the fingertip areas are different.” . In such case, the factors in the authentication classification model should be both fingertip and flight time; and these factors are input to the classification process to create a classification model. Therefore, the next section is to describe the creation processes of the authentication classification model.

3.2.4 Classification and Evaluation

As mentioned in the previous section that the factors to create an authentication classification model can be obtained from statistical testing using non-parametric test because the distributions of observations of all factors are not normal. The creation process uses a freeware names RapidMiner version 9.7 runs on a MacBook Pro, Dual-Core Intel Core i5, memory 8 GB. The first step is to test 1 for all strategy under the Gradient Boost Tree, Random Forest, Decision Tree, and Deep Learning. The results from this step will be evaluated to select the optimized classification model by comparing the accuracy and running process among methods afterwards. Once the method to create the best authentication classification model has been selected using the results from 1 for all strategy, the Scikit-learn is used to create a general model for the authentication classification model.

3.3 The Experimental of the Proposed Text-based CAPTCHA

This section describes the proposed methodology to identify a user based on Text-based CAPTCHA. As mentioned previously, this proposed system is a Text-based CAPTCHA that includes user's keystroke dynamics based on the user's profile data. Figure 10 explains the architecture of the proposed system. According to Figure 10, there are two phases: the user enrollment phase and the user verification phase.

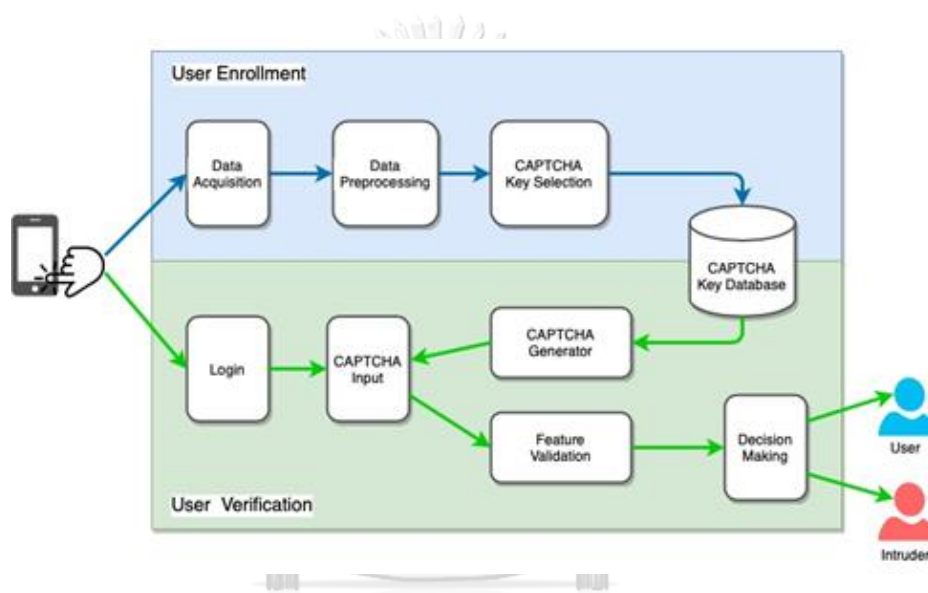


Figure 10 System Architecture for Authentication Classification

Based on Figure 10, the user enrollment phase is the first phase that every user must pass. In this phase, there are 3 modules to be executed: data acquisition, data preprocessing, and CAPTCHA Key Selection. The outcome of the first phase is the individual Text-based CAPTCHA for each user which will be stored in the CAPTCHA_Key database. In the second phase, the user verification phase, this phase consists of 4 modules which are user's log in, CAPTCHA Generator, CAPTCHA input, feature validation, and decision making. The outcome of this phase is the authentication result that indicates the right user or the intruders.

Consider the first phase, at the data acquisition process, the user's data and keystroke information of user's typing rhythm is collected. Then, the data preprocessing module calculates the value of keystroke data such as dwell time, interval time, flight time, latency time, up to up time, and speed of typing. In addition, this module also removes outlier and reducing irrelevant features. Later, the qualified data are frequency counting, ranking, and selecting top 10 of the single character and top 5 of digraph characters. Then, these selected characters are called as the CAPTCHA Keys; and these keys are stored with their related times to the CAPTCHA_Key database.

In the second phase, the user must put the email to the login screen; then the CAPTCHA generator module uses this email to gain access to the CAPTCHA_Key database to generate a profile-based CAPTCHA. Once the profile-based CAPTCHA is presented to the user, the user must key that CAPTCHA while the feature validation module will calculate timing features and compare these times to the correct values in the CAPTCHA_Key database. If the times are matched, the user is approved, otherwise, the user is rejected.



3.3.1 Enrollment phased

3.3.1.1 Data acquisition

In this experiment, the data acquisition module is responsible for capturing information of users that will be used in the authentication process. The user's information is collected via web-based application using iPhone7. Thus, the qualifications of samples in this experiment are volunteers who have good experience in mobile touchscreen typing not less than 3 years and the age range of

the sample is between 18-21 years. According to the COVID-19 problem, the researcher can obtain only 16 volunteers to join the experiment.

Since the web-based application for data entering is very significant and it must be reliable during the data acquisition period, so it is designed and developed using HTML and PHP. In addition, the collecting data is performed by JavaScript. This web-based application works on web browser as Chrome on iPhone7, the screen of collecting information as show as in figure 11. The require data of the user are full name-surname, email, and phone number. Moreover, the data acquisition must be performed only when the data are entered via touchscreen of iPhone7. The functions of this web-based application are features' capturing of keyboard events, such as key press, key down, pressure, or fingertip. These features are vital for finding the keystroke features.

Figure 11 Forms to collect data

Once a sample visits the website via iPhone7, the sample must fill in the demographic information, these are age, gender, education, and occupation. Then, the full name, surname, email, and phone name must be input afterwards. In addition, each sample must repeatedly type for 10 times, as shown in Figure 16.

After the 10th round, the sample presses the confirm button on the webpage, the iOS of iPhone7 will manage the timing of each character, pack and send each character and times to store at the web server.

3.3.1.2 Data preprocessing

When the server receives records from the client, each record contains one character and its key events. The character is stored as a number, called as a key code while the key events are such as key up, key down, and key press. These key events occur whenever the sample presses and releases the key on the virtual keyboard. According to the key events, the keystroke features are calculated and stored as parameters in the CAPTCHA_Key database. Table 4 shows all calculations for keystroke features.

Table 4 The formula for calculating keystroke values

Experiment Feature	Abbreviations	Formula
Dwell time	Dw	$Dw_i = R_i - P_i$
Interval time	In	$In_{(i,i+1)} = P_{i+1} - R_i$
Latency time	La	$La_{(i,i+1)} = R_{i+1} - P_i$
Flight time	Fl	$Fl_{(i,i+1)} = P_{i+1} - P_i$
Up to up time	Up	$Fl_{(i,i+1)} = R_{i+1} - R_i$

*note: i = sequent of character, R=time release, P=time press

Figure 12 shows an example of keystroke features' calculation when 4 keys are pressed, "n", "a", "n", "g". Suppose that the key "n" is pressed at the time (P_1) 75 msec. and "n" is released at the time (R_1) 150 msec. Following by pressing the key (P_2) "a" at the time 175 msec. and release at the time (R_2) 250 msec. According to these data, the dwell time of "n", DW_n , obtains from the difference between 150 and 75, the interval time of "na", In_{na} , obtains from the difference between 175 and 150, latency time of "na", La_{na} , is equal to (250-75), Flight time of "na", Fl_{na} , is the difference between 175 and 75, and Up to up time of "na", Up_{na} , is equal to the difference of (250-150).

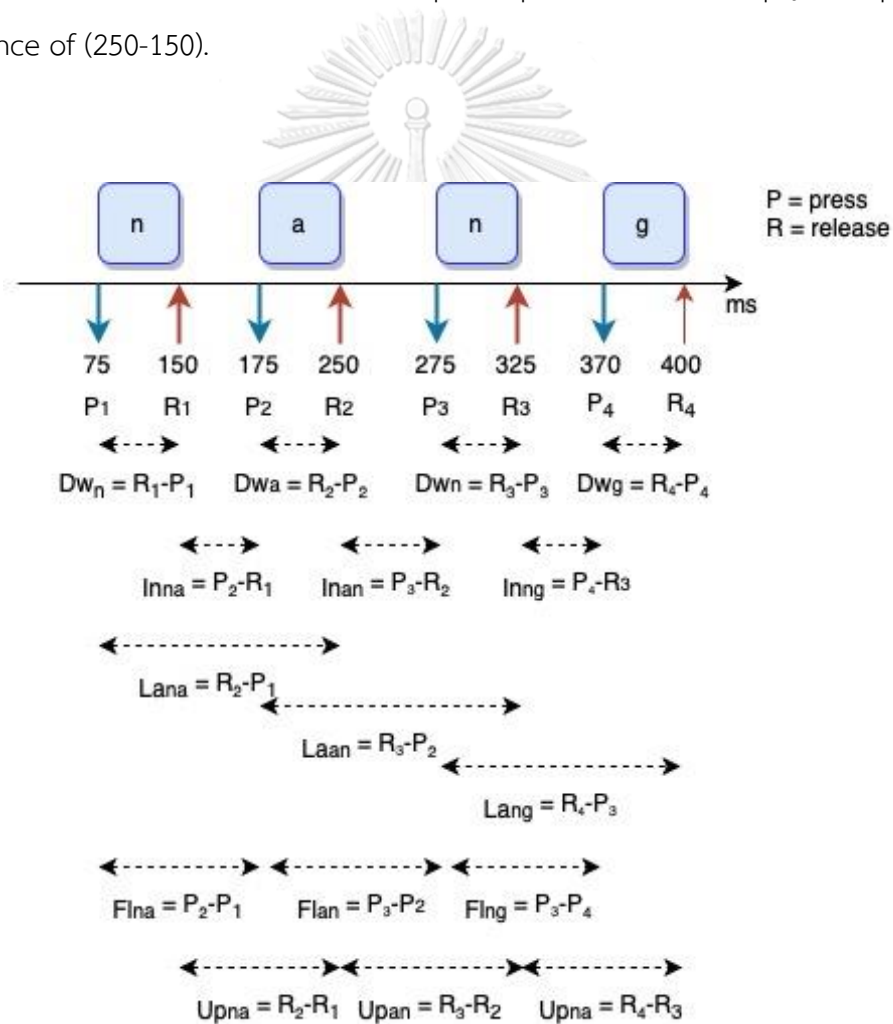


Figure 12 Example of Keystroke features calculation.

After finishing the calculation of keystroke features of every entering character, these data are sent to the CAPTCHA_Key database as shown in Figure 13.

```
{ { "value": "n", "keyUp": 1602766368872, "keyDown": 1602766368779, "offsetX": "20.00",
  "offsetY": "15.00", "pageX": "291.00", "pageY": "625.00", "radius": "25.77" }, { "value": "i",
  "keyUp": 1602766369404, "keyDown": 1602766369279, "offsetX": "25.00", "offsetY": "16.00",
  "pageX": "316.00", "pageY": "530.00", "radius": "38.66" }, { "value": "l", "keyUp":
  1602766369831, "keyDown": 1602766369687, "offsetX": "14.00", "offsetY": "11.00", "pageX":
  "366.00", "pageY": "573.00", "radius": "25.77" }, { "value": "o", "keyUp": 1602766370106,
  "keyDown": 1602766369979, "offsetX": "19.00", "offsetY": "15.00", "pageX": "350.00",
  "pageY": "529.00", "radius": "25.77" }, { "value": "b", "keyUp": 1602766370980, "keyDown":
  1602766370821, "offsetX": "10.00", "offsetY": "31.00", "pageX": "240.00", "pageY": "641.00",
  "radius": "25.77" }, { "value": "o", "keyUp": 1602766371338, "keyDown": 1602766371211,
  "offsetX": "24.00", "offsetY": "14.00", "pageX": "355.00", "pageY": "528.00", "radius": "25.77" },
  { "value": "n", "keyUp": 1602766371663, "keyDown": 1602766371552, "offsetX": "15.00",
  "offsetY": "27.00", "pageX": "286.00", "pageY": "637.00", "radius": "25.77" }, { "value": "space",
  "keyUp": 1602766372007, "keyDown": 1602766371899, "offsetX": "142.00", "offsetY":
  "33.00", "pageX": "254.00", "pageY": "697.00", "radius": "25.77" }, { "value": "n", "keyUp":
  1602766373613, "keyDown": 1602766373478, "offsetX": "20.00", "offsetY": "28.00", "pageX":
  "291.00", "pageY": "638.00", "radius": "25.77" }, { "value": "a", "keyUp": 1602766373837,
  "keyDown": 1602766373709, "offsetX": "10.00", "offsetY": "21.00", "pageX": "37.00", "pageY":
  "583.00", "radius": "25.77" }, { "value": "n", "keyUp": 1602766374046, "keyDown":
  1602766373926, "offsetX": "23.00", "offsetY": "26.00", "pageX": "294.00", "pageY": "636.00",
  "radius": "25.77" }, { "value": "g", "keyUp": 1602766374337, "keyDown": 1602766374254,
  "offsetX": "27.00", "offsetY": "20.00", "pageX": "217.00", "pageY": "582.00", "radius": "38.66" },
  { "value": "l", "keyUp": 1602766374696, "keyDown": 1602766374577, "offsetX": "8.00",
  "offsetY": "19.00", "pageX": "360.00", "pageY": "581.00", "radius": "25.77" }, { "value": "a",
  "keyUp": 1602766374921, "keyDown": 1602766374793, "offsetX": "8.00", "offsetY": "28.00",
  "pageX": "35.00", "pageY": "590.00", "radius": "25.77" }, { "value": "e", "keyUp":
  1602766375413, "keyDown": 1602766375320, "offsetX": "29.00", "offsetY": "17.00", "pageX":
  "117.00", "pageY": "531.00", "radius": "38.66" } }
```

Figure 13 Example of the streaming transaction from a client to the CAPTCHA_Key database.

Outlier detection

Once all entering data are sent to the CAPTCHA_Key database, these data must be validated for their suitability before sending to the CAPTCHA key selection process. The suitability of the data is based on the consistency of all data, such as time values that may be affected from some external factors while the sample was typing. Consequently, the outlier must be removed using Empirical rule that 95% of population will fall in the range of $(\mu \pm 2\sigma)$, as shown in Figure 14, the value of μ and σ can be obtained from the Equation 2 and 3.

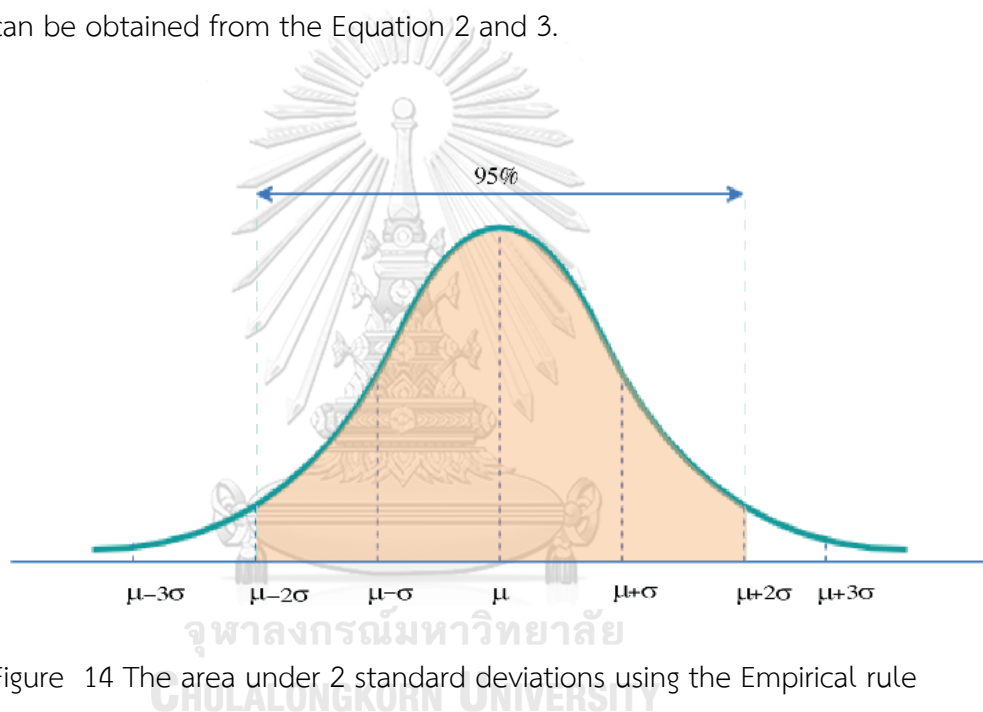


Figure 14 The area under 2 standard deviations using the Empirical rule

Where:

$$\mu = \frac{\sum_{i=1}^n x_i}{N} \quad \text{Equation 2}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad \text{Equation 3}$$

3.3.1.3 CAPTCHA Key Selection

After cleaning the data in the database, the counting for arranging the rank of characters and digraph is performed. Thus, the results of this counting are frequency of each entered character, and frequency of each digraph or frequency of a group of two successive letters. For example, the input data is “nang” then, the single characters are “n”, “a”, “n”, and “g”, and the digraph characters are “na”, “an”, and “ng”; the frequency of “n” is 2, “a” and “g” is 1, like “a” and “g”, “na”, “an”, and “ng” have frequency as 1.

Once the list of frequency counting character is presented, the CAPTCHA key selection module ranks the frequencies of characters from the most typed to least typed to select the top ten of the single character and top five of digraph characters. These selections are the CAPTCHA keys. Finally, these CAPTCHA keys are stored in the CAPCHA_Key database.

3.3.2 User Verification

The user verification phase is the phase when the user wants to access the system. The objective of this phase is to authenticate the user before authorizing the user to access the system. Within this verification process, there are 5 modules to be executed as shown in Figure 16. Details of each module are elaborated as follow.

a) Login

Login is the first module in the User Verification phase that the user must use the user's email for login to the system. This email of the user is used to match the user's email in the CAPTCHA_Key database to generate the Text-based CAPTCHA that suitable to the email's owner. The module that is responsible for generating the specific CAPTCHA is the CAPTCHA Generator, details are described as follow.

2) CAPTCHA Generator

Based on the entering email, this input data will randomly select the CAPTCHA key from the CAPTCHA_Key database to create an individual Text-based CAPTCHA for the email owner. Since the CAPTCHA_Key database stores a set of CAPTCHA keys for each user that consists of 15 strings: 10 single characters and 5 digraph characters, this set of CAPTCHA keys will be randomly selected to generate an individual Text-based CAPTCHA. For randomly generating Text-based CAPTCHA, a series of 6 characters is illogically chosen from the defining character set, that have no pattern. With the utility of generating a random string, the output is inimitable of each sample. Consequently, the security of CAPTCHAs can be obtained, and individual Text-based CAPTCHA can be achieved.

The probability for making a copy of the individual Text-based CAPTCHA can be demonstrated as follow.

For each set of characters that contains 10 single characters and 5 digraph characters, the member of this set is counting as 15; each digraph is counted as one character. Thus, there are 15 possible values to be in each digit of the CAPTCHA length of 6. Thus, all possible values for 1 CAPTCHA with 6 positions can be 15^6 ; this value is equal to 11,390,625. Since only 1 CAPTCHA will be chosen from this sample space, then the probability to guess the right Text-based CAPTCHA is $\frac{1}{11390625} \approx 0.000877$. The probability is close to zero, this mean it is difficult to intruder to guess and attach this Text-based CAPTCHA. For the security reasons, random generator can be useful. Moreover, the security of the CAPTCHA is varied by the number of digits used.

The procedure of CAPTCHA generator:

```

Pseudo code of CAPTCHA generator

  READ characterSet from CAPTCHA_key database

  INIT captchaLength to 6

  SET count to 0

  SET captcha to EMPTY STRING

  WHILE (count < captchaLength)

    SET randomChar by pick one of a character from characterSet randomly

    SET captcha to captcha + randomChar

    INCREMENT count

  END

  Print captcha
  
```

The result from running the CAPTCHA generator can be displayed as examples in Figure 15.

rthros	hosrts
toh1rr	at1a15
ettoatr	111oho
tar16ar5	at5abtao
ratt1h6	btt5ro5

Figure 15 Samples of 6 digits of the Text-based CAPTCHA generated by the CAPTCHA Generator

3) CAPTCHA Input

After the CAPTCHA Generator generates a Text-based CAPTCHA from data retrieved from the CAPTCHA_Key Generator database, a randomly created Text-based CAPTCHA will be created without asking for more information. Then, the user just entered the presented Text-based CAPTCHA via the CAPTCHA Input screen. During the entering period, the process of feature validation starts.

4) Feature Validation

Once the user types each character of the Text-based CAPTCHA, the Feature Validation module will response for finding the lower and upper limits using the boundary of the Empirical rule as $\bar{x} \pm 2\sigma$. So, if there is a 6-digit CAPTCHA, “r t h r o s”, the user must enter 6 characters as appeared. Therefore, each entering character is validated by comparing its time, the Input time value, with the interval of the character-profile’s time stored in the CAPTCHA_Key Generator database, $(\bar{x} - 2\sigma, \bar{x} + 2\sigma)$, as shown in Table 5. After every character has been validated its typing time, the results will be sent to the Decision-making module for the final authentication result.

Table 5 Samples of time comparison between the input time with the valid time domain.

Character	Input time value	$\bar{x} - 2\sigma$	$\bar{x} + 2\sigma$	output
r	101	97.2491905	184.75081	Valid
t	78	46.9127635	138.053903	Valid
h	119	94.0367696	184.16323	valid
r	101	97.2491905	184.75081	Valid
o	102	73.7626938	182.477306	Valid
s	111	74.9559307	180.444069	Valid

5) Decision-making

After each character is individually validated, the results are input to classify between the real user and imposters. The expected outcome is the determination of the authenticated user. This experiment applies the Gradient Boost Trees (GBTs) technique, a machine learning technique for regression and classification of the decision tree model, for classifying the user using keystroke dynamics feature. The core structure of GBTs is produced from the weak model of decision trees by combining many learning trees to build the final model that is much accurate.

The algorithm of the GBTs is recurring create a strong classification decision tree model from the weak model; the descendent GBTs is the error reduction from the

previous GBTs. So, the performance of the algorithm is high efficiency on both regression and classification tasks and higher secure than other trees. Figure 16 demonstrates how GBTs was created and the error rate decreases while the number of trees in model is raising.

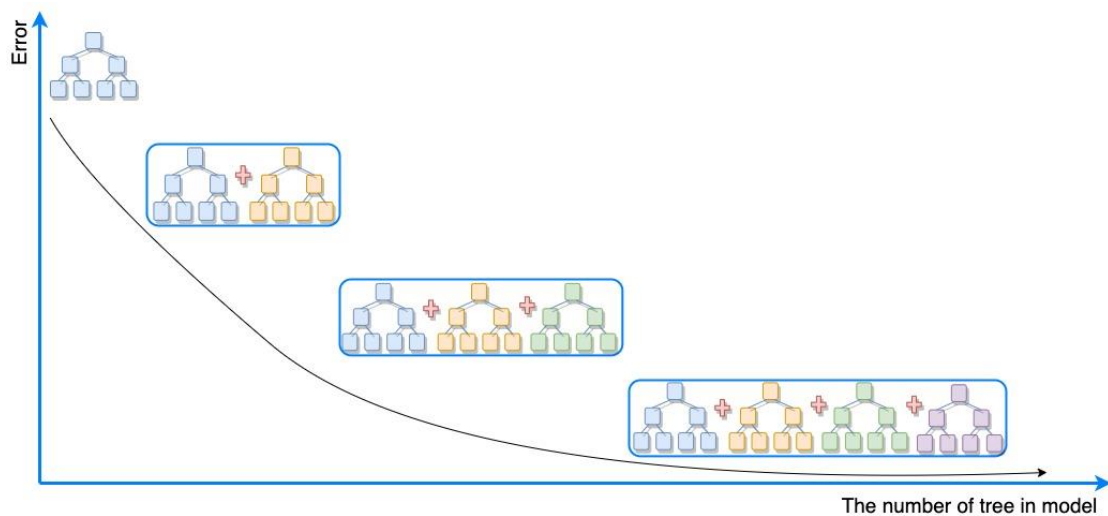


Figure 16 A simple example of visualization of Gradient Boosted Trees

3.3.3 Performance evaluation

To confirm the proposed method of producing a strong Text-based CAPTCHA that suitable for each user, the performance of the proposed mechanism must be determined. Therefore, this research performs two phases of the evaluations. The first phase is to simulate a situation that a user types the user's Text-based CAPTCHA then others are trying to break his typing afterwards. The second phase is to use the bot to attack the user's Text-based CAPTCHA. Details of each evaluation method are described below.

1) First evaluation phase

This phase uses two randomly generated Text-based CAPTCHAs obtained from the CAPTCHA generator; these Text-based CAPTCHAs are “r t h r o s” and “e t t o a t r”. The difference between “r t h r o s” and “e t t o a t r” is related to the types of characters used when creating the word. For “r t h r o s” is created from the single character stored in the CAPTCHA_Key database while “e t t o a t r” is created by combining from the single character and digraph character.

Using a machine learning based on Gradient Boosted Tree running on RapidMiner v6 for evaluating the proposed method in creating a personal Text-based CAPTCHA that can accept the real user and reject the intruder. The cross validation in RapidMiner is chosen to take care the creation and the process splits into 10 folds for both training, and testing data.

In the first round, the data of a user was selected and split into 2 datasets, those are training set, 70%, and testing set, 30%. With this dataset, a Gradient Boosted Trees was created and tested as perfect model for a particular user. Then, the 30% of another user was selected for testing the model. If the Gradient Boosted Trees obtained in the first round is perfect, then the testing data from another user will not be able to break with high accuracy of the classification.

All processes in the first evaluation phase are displayed in Figure 17.

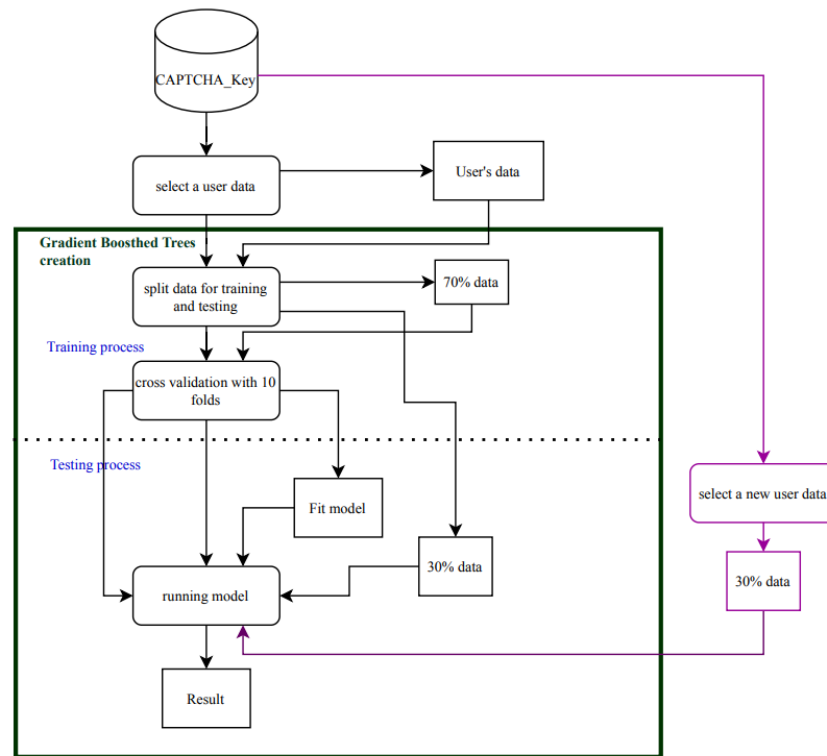


Figure 17 The evaluation process in the first phase.

The performance indexes for this classification model are accuracy, precision, and recall. The accuracy measures the testing results by the percentage of correctly classified. Precision and recall are the evaluation of the efficiency of the model. So, those metrics can be calculated as the follows.

$$accuracy = \frac{\sum a+b}{\sum a+b+c+d} \quad \text{Equation 4}$$

$$precision = \frac{\sum a}{\sum a+c} \quad \text{Equation 5}$$

$$recall = \frac{\sum a}{\sum a+d} \quad \text{Equation 6}$$

Each variable in the formula has meaning as, a is the number of true positive; b is the number of true negative; c is the number of false positive; d is the number of false negative.

2) Second evaluation phase

Although the first evaluation phase confirms the efficiency of the invented Text-based CAPTCHA in the human's attack, the second phase is executed to confirm the efficiency for bot's attack. Thus, this testing phase runs two simulators to beat the individual Text-based CAPTCHA. The first simulator uses the brute force method to break the Text-based CAPTCHA obtained from the CAPTCHA generator module. Moreover, this simulator is an online system from https://tmedweb.tulane.edu/content_open/bfcalc.php?uc=&lc=&nu=&sc=&ran=6&ran_s=&dict=. The ability of this online bot is the attempt to guess the set of 6-digit characters, and 7-digit characters.

Except the online simulator mentioned above, there are 2 more self-written bots to defeat the individual Text-based CAPTCHA. The first one is Typing CAPTCHA Simulator Bot; this bot runs as known-CAPTCHA attack. Another bot is Typing CAPTCHA Time Delay Simulator Bot; this bot is similar to the first bot except it includes counting delay within the attacking routine. The pseudocodes to solve this CAPTCHA are as the follow.

The pseudocode of Typing CAPTCHA Simulator Bot, the malware starts with reading the characters in CAPTCHA wording and then type the characters. After the malware clicks for submission the set of character to the system. Finally, the system returns the results. The pseudocode to solve this CAPTCHA are as the follow.

Typing CAPTCHA Simulator Bot:

```

#TypingCAPTCHA Simulator Bot

READ captchaWording

Go to login with captcha page

FOR iteration over character of captchaWording
    Type character
ENDFOR

Click validate button in the captcha page

GET the validate result

SET validate result to validatedResult

PRINT validatedResult

```

The pseudocode of Typing CAPTCHA with special condition, that start with the following steps.

- Choose gender for simulator bots
- Checking finger pressure
- Checking fingertip
- Checking typing time of each character of Text-based CAPTCHA that appear to type.

The procedure of this simulator chooses gender for simulator bot. If the chosen gender is matched with the real gender, the bot starts to generate the finger pressure with 0.001 unit following with a set of data at 0 millisecond into the field of CAPTCHA. If the CAPTCHA system returns false, the malware repeats sending a new set of data to the field of CAPTCHA after increasing 1 millisecond to the typing time, and 0.001 unit to the pressure until the outcome is true. This means the CAPTCHA system has been broken. For other factors, such as latency time, interval time, dwell time, flight time, and up to up time, those check when the typing value returns true. The pseudocodes to solve this CAPTCHA are as follow.



Typing CAPTCHA Simulator Bot with special condition:

```

# Find the correct pressure, delay and gender by automation BOT
SET isSuccess to FALSE

SET factor to findCorrectFactor(male)

IF factor = NULL THEN
    SET factor to findCorrectFactor(female)
END IF

IF factor = NULL THEN
    PRINT "Factor not found"
ELSE
    PRINT factor.correctGender
    PRINT factor.correctDelay
    PRINT factor.correctPressure
END IF

FUNCTION findCorrectFactor(correctGender)
    READ captchaWording

    INIT delayIncrementalStep to 1
    INIT maximumMillisDelay to 1000

    INIT pressureIncrementStep to 0.001
    INIT maximumPressure to 1

    INIT fingertipIncrementStep to 1
    INIT maximumFingertip to 100
  
```

SET validatedResult to FALSE

SET correctDelay to 0

SET correctPressure to 0

SET correctFingertip to 0

FOR pressure = 0 to maximumPressure

FOR delayMillis = 0 to maximumMillisDelay

FOR fingertip = 0 to maximumFingertip

Go to login page

FOR iteration over character of captchaWording

Type character with holding a delay equal to

delayMillis

ENDFOR

ENDFOR



Click validate button in the captcha page

GET the validate result

SET validate result to validatedResult

IF validatedResult = FALSE THEN

SET delayMillis to (delayMillis +
delayIncrementalStep)

ELSE

```
        SET correctDelay to delayMillis
    END IF

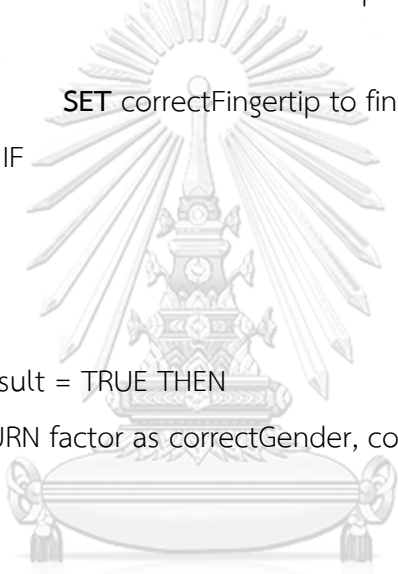
END FOR

IF validatedResult = FALSE THEN
    SET pressure to (pressure + pressureIncrementStep)
ELSE
    SET correctPressure to pressure
ELSE
    SET correctFingertip to fingertip
END IF

END FOR

IF validatedResult = TRUE THEN
    RETURN factor as correctGender, correctDelay, correctPressure and
correctFingertip
ELSE
    RETURN NULL
END IF

End FUNCTION
```



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Chapter 4

Results

This section shows the results of the evaluation processes that have been mentioned in the previous section.

4.1 Results of factor analysis

This section shows the result from the data collection using volunteer sampling and the sample group was obtained from 3rd year undergraduate students from Computer Science program, Faculty of Science, Chulalongkorn University. The size of the sample is 18 students, 8 males and 8 females. In addition, these samples are in the sitting position while they input data via smartphone's touchscreen. Later, these data are input to prove the hypothesized using 95% confident level as follow.

To perform further statistical test, all data distributions must be confirmed as they are normal. Therefore, after normality testing using Kolmogorov-Smirnov for the normal distribution of every parameter, the result is rejected since the *p-value* is equal to $0.00 < 0.05 = \alpha$, for all factors. Thus, it can conclude that the distributions of all collected data are not normal, and the non-parametric test is applied. So, in the case of testing for the gender's effects, the Mann-Whitney U test is deployed, and the test for the fingertip's effect is Kruskal-Wallis test. Details of all tests are described as follow.

4.1.1 Mann-Whitney U test: Test for Gender's effect

According to all hypothesizes those are setting in the previous section, the use of Mann-Whitney U test under the significant level of 0.05 can show that male and female have significant differences in all mean values of times which are dwell time, interval time, latency time, flight time, and up-to-up time, with $p\text{-value} = 0.00 < 0.05 = \alpha$. These results indicate that when combining gender with any keystroke features can be factors to distinguish a person from others.

4.1.2 Kruskal-Wallis test: Test for Fingertip's effect

Based on the hypothesizes in fingertip's effect testing, the non-parametric method that is applied to prove such cases is Kruskal-Wallis test as it is used for n -independent samples. The results of these test can identify that different sizes of fingertips, which varies from 1 to 7, can have at least one significant impact to the mean values of some keystroke features under the confident level of 95%. For example, when testing the impact of fingertips towards the value of dwell times, it may or may not cause the dissimilarity of dwell times when the sizes of the fingertips are dissimilar. Therefore, after testing with Kruskal-Wallis using 5% significant level, it can signify that the fingertips can influence the mean values of dwell time, latency time, flight time, and up-to-up time, including the mean of the finger pressure, because the p -values of all tests are equal to zero which is less than the significant level.

In accordance with the results from both tests, gender's effect and fingertip's effect, it can predict that the personality's determinator are gender, fingertip, finger pressure, dwell time, flight time, interval time, latency time, and up-to-up time. Therefore, the next step is to confirm the suitability of 7 factors in the machine

learning models using 1 to all strategy, running RapidMiner version 9.7. The measurement metrics of the model suitability are accuracy, precision, recall, and execution time. Details of the factors' suitability testing of 7 factors are drawn as follow.

4.1.3 Factors Suitability Testing

Based on the results of the previous process, the factors those are counted as parameters for the authentication classification model for each sample are gender, fingertip, finger pressure, dwell time, flight time, interval time, latency time, and up-to-up time. Therefore, these factors will be input to RapidMiner version 9.7. Since there are numerous classification models to be chosen in RapidMiner version 9.7, this research selects 1 for all strategy with 4 significant models: Gradient Boost Tree, Random Forest, Decision Tree, and Deep Learning. Moreover, all 16 samples were run as input data of the RapidMiner under the 1 to all strategy, and the performance metrics in this process are accuracy, precision, recall, and execution time. Therefore, one classification model is derived for one sample. Finally, there are 16 individual classification models for 16 individual samples. Since this research tried on 4 dissimilar models, each measurement value is calculated individually. However, when consider all 4 models of all 16 samples, the measurement metrics of these model are higher than 95% and some models of some samples are high up to 100%. Figure 7 shows the stack graph of the accuracy values of 16 samples when uses 4 different classification models: Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees. This graph shows that the accuracy values of all 16 samples are closed to 100% for every model. Moreover, the precision values of all 16 samples using 4 classification models can be demonstrated in Figure 18.

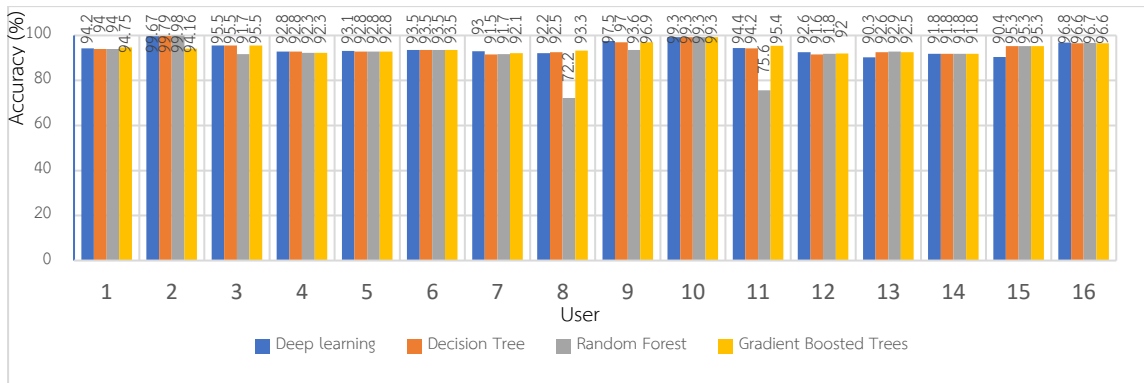


Figure 18 The accuracy values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.

According to all precision values in Figure 19, every model has precision value is more than 90% which can be interpreted that using either one of the four models with the proposed factors can precisely identify the imposter.

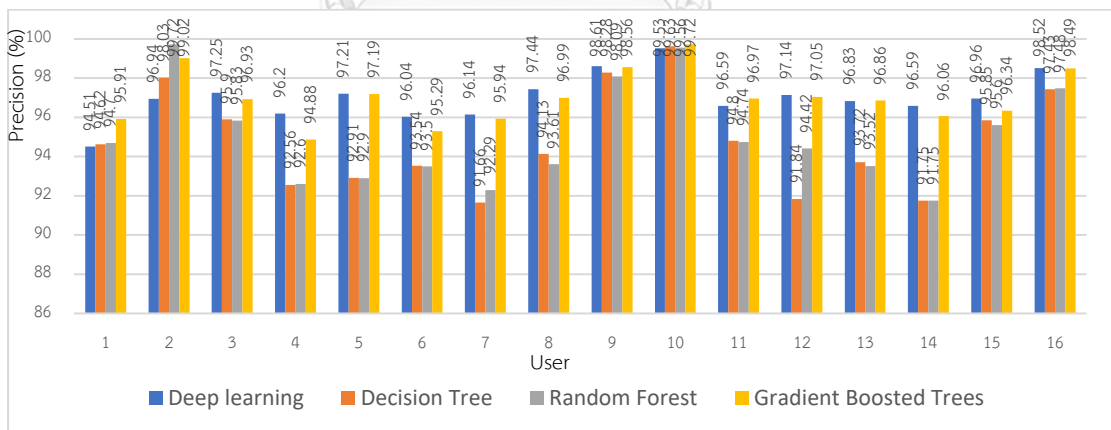


Figure 19 The precision values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.

Like the precision values in Figure 20, the recall values in Figure 20 indicates that the most of 4 models are close to 100% except Deep learning is less than other a little. These values are consistent with the accuracy and precision presented previously.

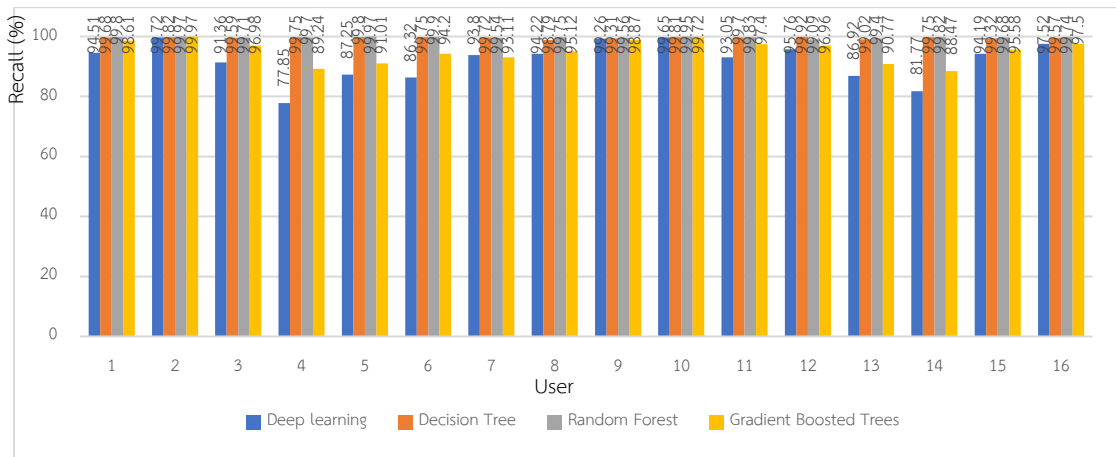


Figure 20 The recall values of Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.

Even though the first three metrics provide positive measurement values for the proposed factors in all 4 classification models, the last metrics, the execution times, does the opposite way. As shown in Figure 21, the execution time of the Random Forest is the highest among four methods while the execution time of the Decision Tree is the smallest. Thus, the Random Forest might not be suitable for being a classification model although it has high accuracy, precision, and recall values.

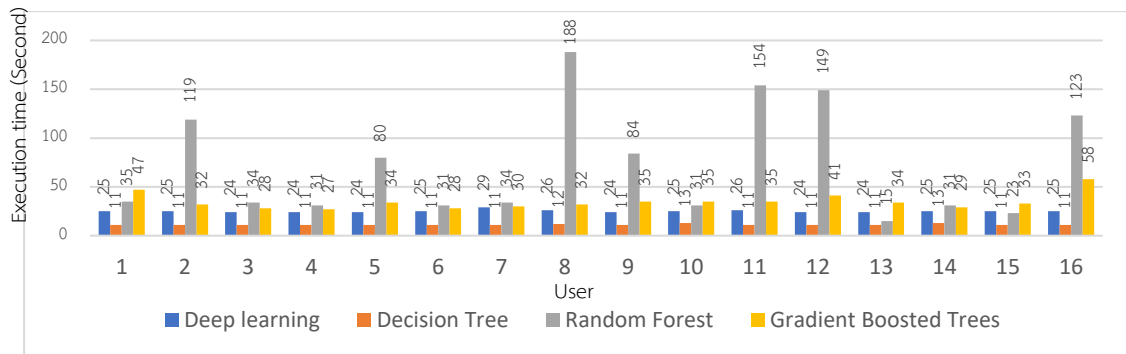


Figure 21 The different execution times using Deep Learning, Decision Tree, Random Forest, and Gradient Boosted Trees.

Even though the first three metrics provide positive measurement values for the proposed factors in all 4 classification models, the last metrics, the execution times, does in the opposite way. As shown in Figure 21, the execution time of the Random Forest is the highest among four methods while the execution time of the Decision Tree is the smallest. Thus, the Random Forest might not be suitable for being a classification model although it has high accuracy, precision, and recall values. Therefore, the proposed 7 factors, which are gender, fingertip, finger pressure, dwell time, flight time, interval time, latency time, and up-to-up time, can be used as the authentication classification factors under either one of these three models: Deep Learning, Decision Tree, or Gradient Boosted Trees. However, the Gradient Boosted Trees [84] is a learner algorithm that used the weak learners into the strong learners, that mean the weak learner attempts to minimize the error of the previous tree to make the next tree is highly efficient model, so the strong learner has an almost perfect performance. The model is a machine learning technique for regression and classification problem to predict model. This approach produces a predict model in form of a sequence of Decision Tree, while each Decision Tree improves accuracy from the error of the previous trees. Therefore, the recommended models are either Deep Learning or Gradient Boosted Trees. Thus, the averages of all metrics are calculated and shown in Table 6.

Gradient Boosted Trees is a learner algorithm that used the weak learners into the strong learners, that mean the weak learner attempts to minimize the error of the previous tree to make the next tree is highly efficient model, so the strong learner has an almost perfect performance. The model is a machine learning technique for regression and classification problem to predict model. This approach produces a predict model in form of a sequence of Decision Tree, while each Decision Tree improves accuracy from the error of the previous tree.

Table 6 Averages of all metrics

Features	Deep Learning	Gradient Boosted Trees
Accuracy	94.19%	94.26%
Precision	97.03%	97.01%
Recall	92.07%	98.21%
Execution Time (second)	25	35

From the values presented in Table 6, the accuracy and the recall values of Gradient Boosted Trees is higher than Deep Learning. Although the execution time of the Gradient Boosted Trees is about 35 milliseconds, higher than the Deep Learning, but this value is the average of 1 to all attackers. Therefore, time for one attack detection will be much smaller. So, by conclusion, the suitable authentication classification model should be the Gradient Boosted Trees.

4.1.4 Factor weight

From previous section the gradient boosted tree algorithm is suitable to be authentication model, so in this section determine the weight factors of 8 factors as show as in Table 7.

Table 7 The weight factors of 8 factors

Factor	Average weight
Gender	0.2529
Pressure	0.1308
Latency	0.1184
Interval	0.1108
Dwell	0.1073
Fingertip	0.0971
Flight	0.0785
Up to up	0.0729

The results in table 7 shows that the weight factor work in classification model. However, the maximum weight factor is Gender, that is effective in identification process.

4.1.5 Optimization factors

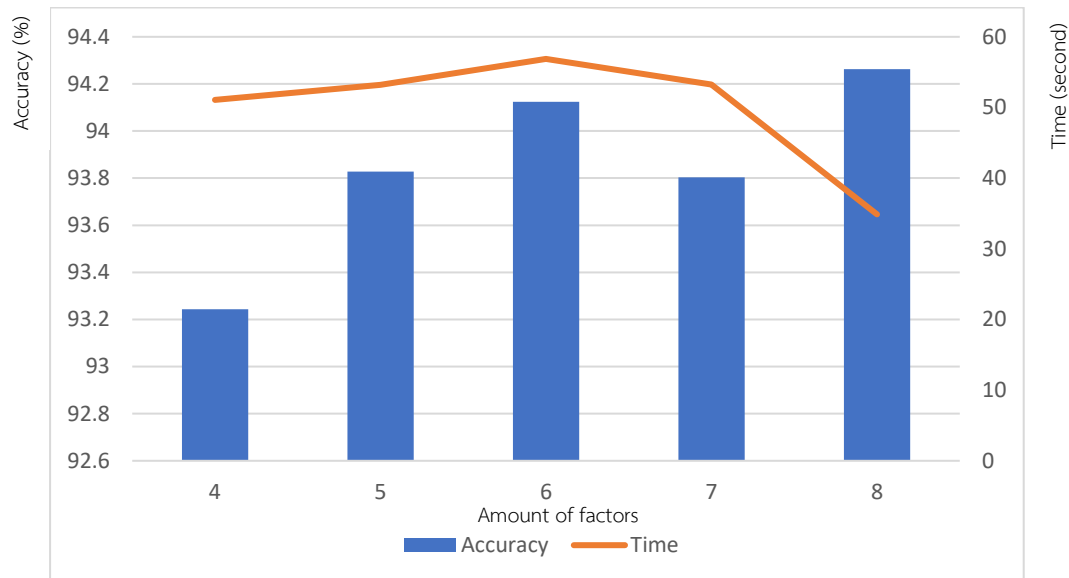


Figure 22 The accuracy that cutting the factors

From the figure 22 shows the accuracy using eight factors the accuracy is high as 94.3%, and the time execute is 32 second. Decreasing only, one factors as up to up time, the accuracy is 93.8%, and the time execute is about 50 second. Decreasing two factors as up to up time and flight time, the accuracy is 94.1%, and the time execute is about 55 second. Decreasing three factors as up to up time, flight time, and fingertip, the accuracy is 93.83%, and the time execute is about 50 second. And then decreasing four factors as up to up time, flight time, fingertip, and dwell time, the accuracy is 94.1%, and the time execute is about 22 second. So, the accuracy of 8 factors is higher than decreasing the factors, and with the time is lowest than the other.

4.2 The consideration of typing characteristics in various conditions

Case 1: Comparison of the typing characters between a legitimate user and intruder

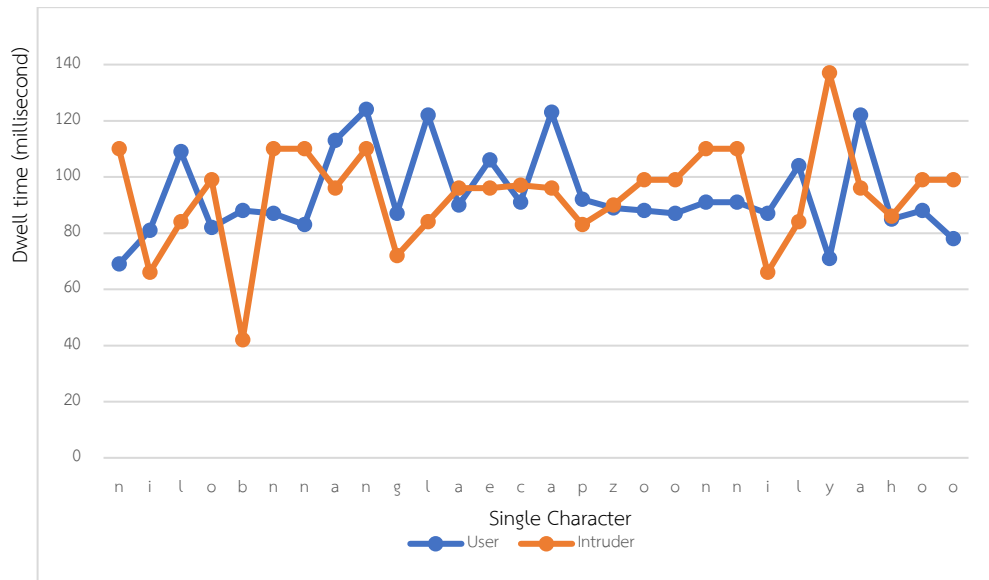


Figure 23 comparison of the individual typing character between a legitimate user and intruders

Figure 23 shows the dwell time of the individual typing characters of a legitimate user and intruders. This line graph indicates that every intruders' typing character is different from the legitimate user. So, it can be claimed that any legitimate users have unique typing time of their own full-name and surname which all intruders cannot emulate to the legitimate user.

Case 2: Times of all typing di-graph characters of the legitimate user

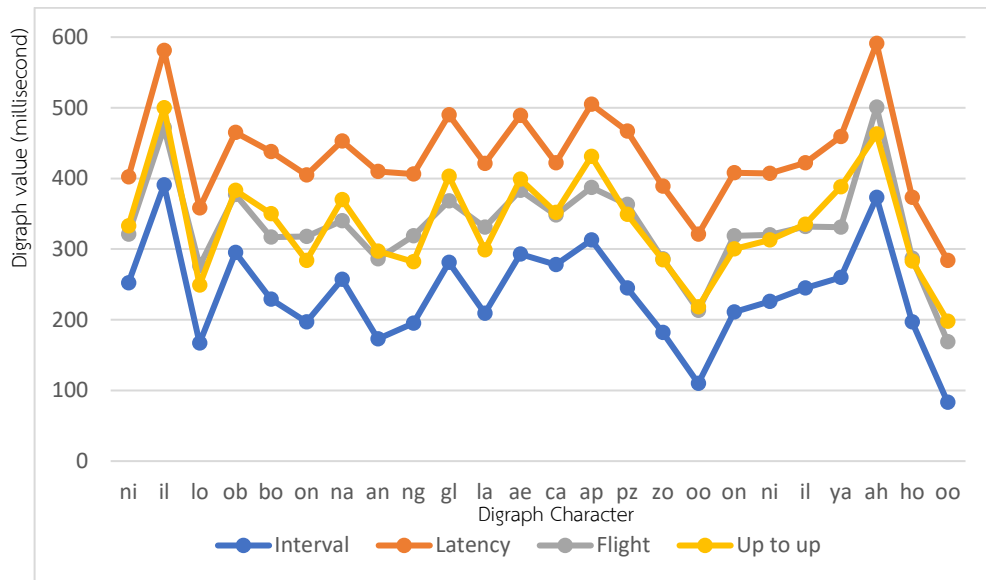


Figure 24 Display all typing times of di-graph characters of a legitimate user

Figure 24 shows all typing times of di-graph based on a legitimate user. These lines indicate the differences of times according to the di-graph characters. Moreover, the typing pattern of the user can be clearly noticeable.

Case 3: Legitimate user types individual Text-based CAPTCHA

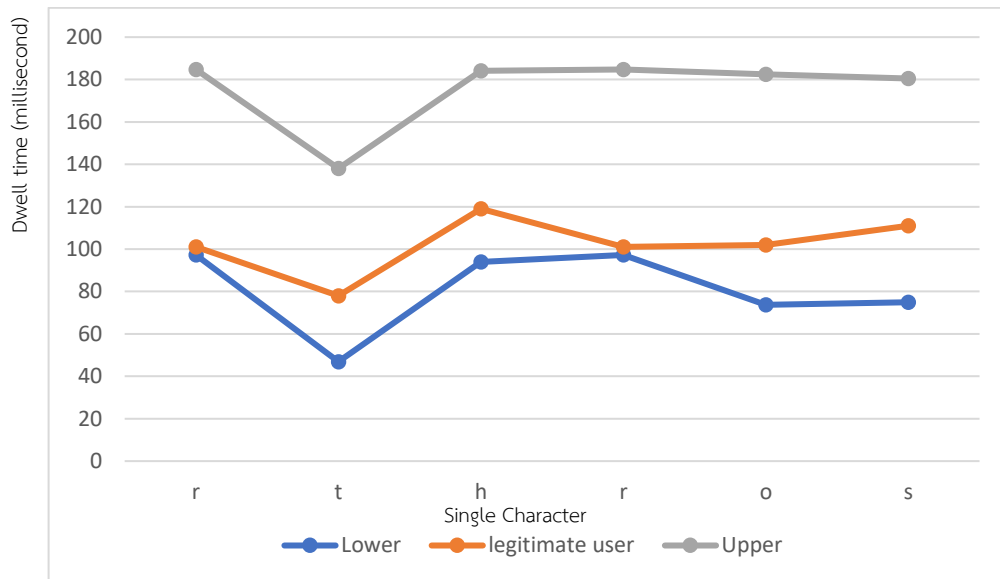


Figure 25 The line graph of dwell times based on an individual Text-based CAPTCHA typing of a legitimate user.

Figure 25 shows all typing of CAPTCHA as “r t h r o s” based on a legitimate user. These lines indicate the differences of time according to single characters of the individual Text-based CAPTCHA. The line graph indicates that the legitimate user can typing in each character’s boundary.

Case 4: Legitimate user types individual Text-based CAPTCHA that include digraph characters

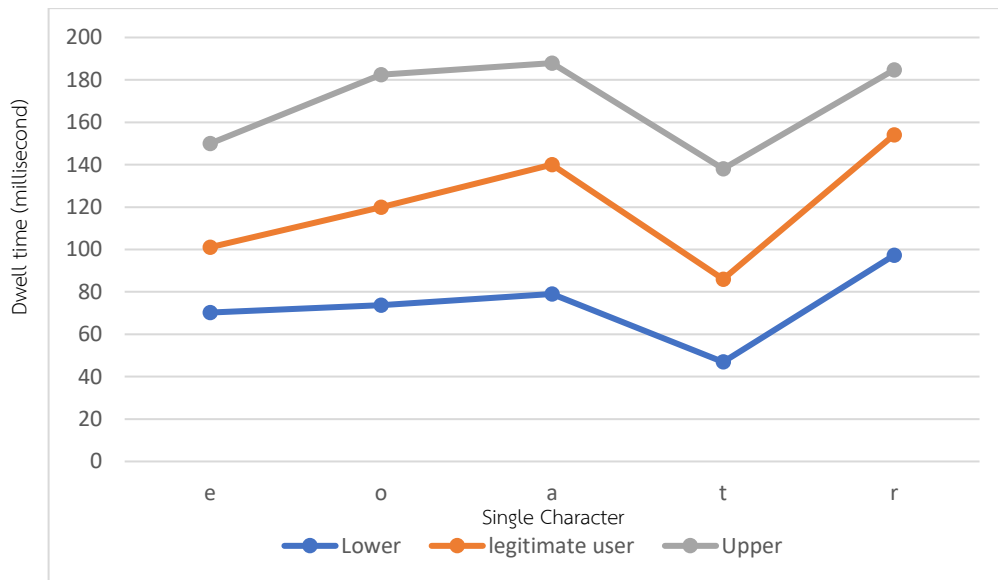


Figure 26 The line graph of dwell and digraph times based on an individual Text-based CAPTCHA typing of a legitimate user.

Figure 26 shows dwell times when the legitimate user types the generated CAPTCHA that contains digraph characters, “e tt o a t r” where the lines clearly determine that the dwell times of each character, except digraph characters, of the user are still in the character’s boundary. Moreover, all times of the digraph “tt’ are displayed in Figure 27 with their boundaries

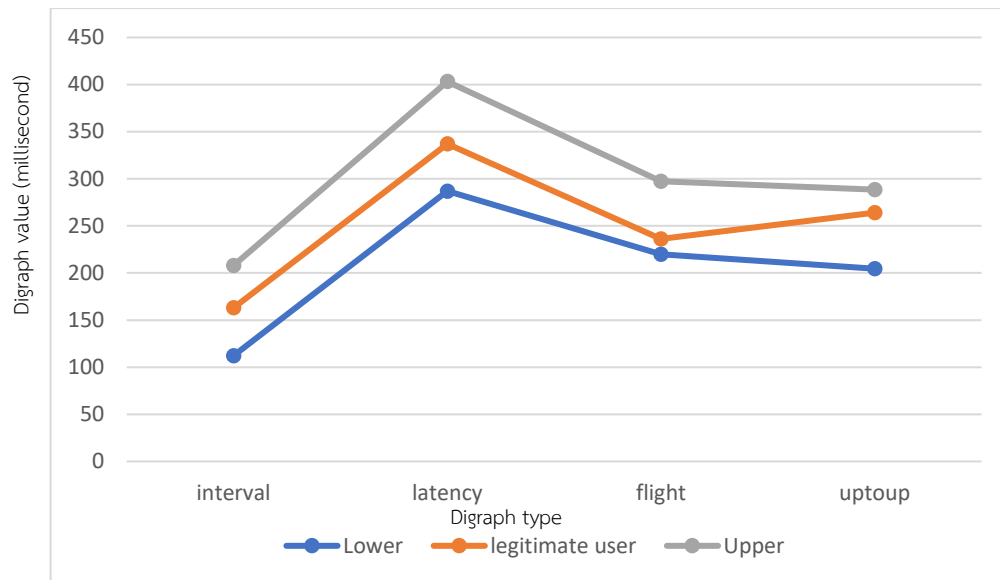


Figure 27 Legitimate user types individual Text-based CAPTCHA that include digraph character (Digraph only)



Case 5: Intruder types the individual Text-based CAPTCHA

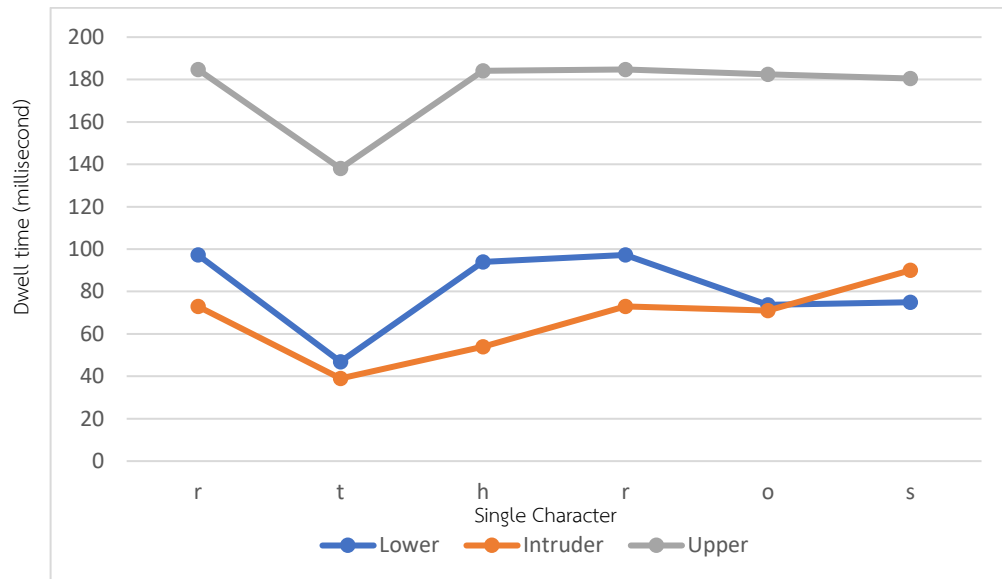


Figure 28 The line graph of dwell times based on a derived Text-based CAPTCHA typing by an intruder.

Figure 28 shows all dwell times of CAPTCHA, “r t h r o s”, that typed by an intruder. This line graph indicates that there are differences in the typing times of the intruders when comparing with the character’s boundary of the legitimate user.

Case 6: Intruder types an individual Text-based CAPTCHA with digraph characters

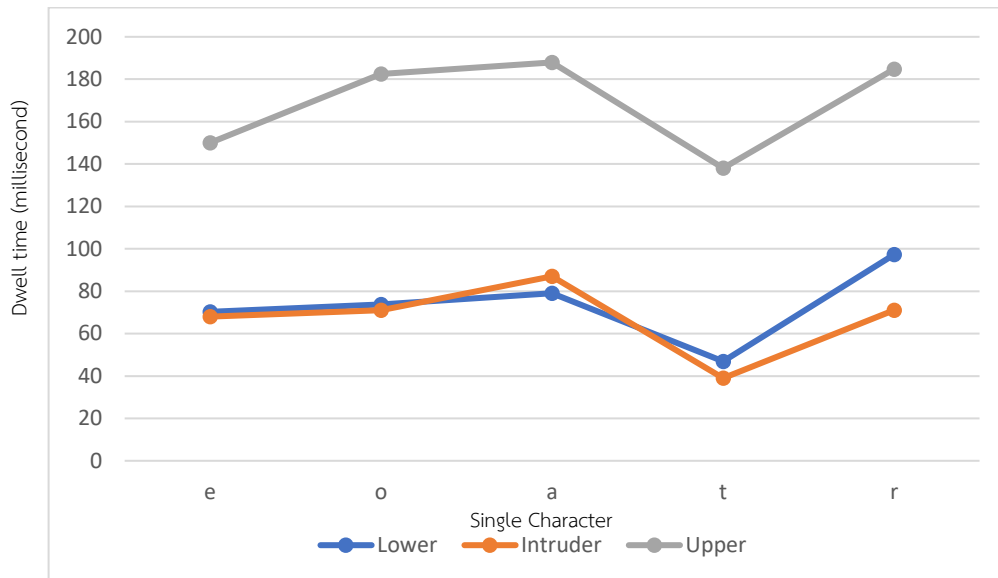


Figure 29 The line graph of dwell times for each character of an individual Text-based CAPTCHA typing by a intruder.

Figure 29 shows all typing of CAPTCHA as “e tt o a t r” based on a intruder. These lines indicate the differences of time according to single characters of the individual Text-based CAPTCHA. The line graph indicates that the intruder cannot completely type within all characters’ boundaries. In addition, the typing times of digraph based on an intruder which are shown in Figure 30 also confirm that the intruder cannot type within the same times’ boundaries of the legitimate user.

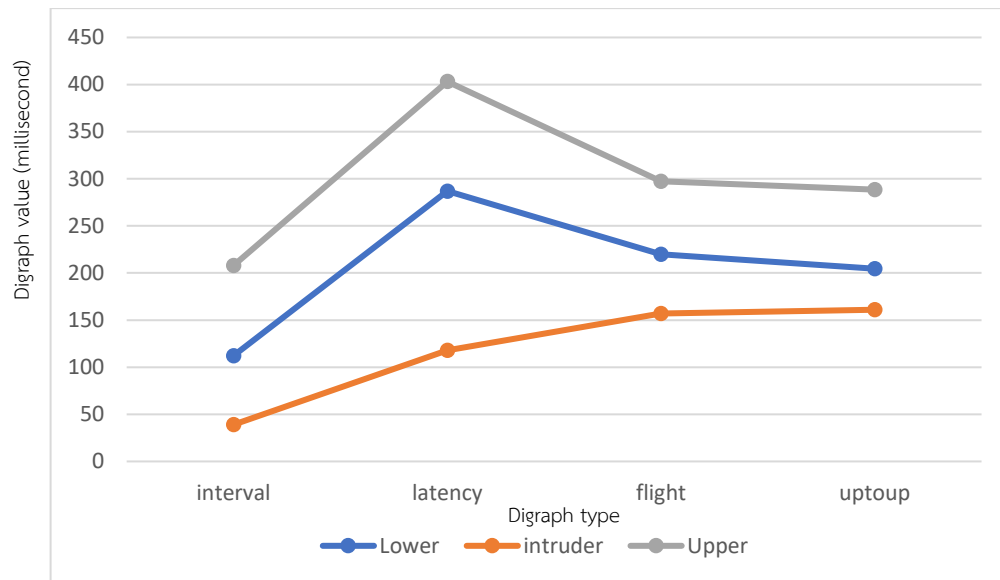


Figure 30 Intruder types individual Text-based CAPTCHA with digraph characters
(Digraph only)



4.3 The result of the first evaluation

Since the first phase uses Gradient Boosted Trees algorithm to evaluate the performance of the proposed Text-based CAPTCHA, the results of two test states, as shown in Figure 17, are shown in Table 8 and Table 9.

Table 8 The result of the real user tries to enter the Text-based CAPTCHA as “r t h r o s” and “e t t o a t r”

	“r t h r o s”	“e t t o a t r”
Accuracy	100%	100%
Recall	100%	100%
Precision	100%	100%

According to the performance measurement values in Table 8, 100% of accuracy, 100% of recall, and 100% of precision, it means the Gradient Boosted Trees algorithm can 100% detect the right user whenever the user enters the system with the generated Text-based CAPTCHA.

Table 9 The result of the other user tries to enter the Text-based CAPTCHA as “r t h r o s” and “e t t o a t r”

	“r t h r o s”	“e t t o a t r”
Accuracy	59.43%	59.79%
Recall	55.44%	45.09%
Precision	41.76%	42.84%

As the test in state 2, the entering times of another sample were used to emulate as the entering times of the owner, the results show that the Gradient Boosted Trees algorithm can detect as if it was the real user with 59.79% accuracy, 45.09% of recall, and 42.84% of precision. Thus, there are high possibility that the authentication system will reject the intruders.

4.4 The result of the second evaluation

4.4.1 Brute force online simulator

Since the derived Text-based CAPTCHA has length of 6 or 7 characters, the second evaluation for the strength of the Text-based CAPTCHA with these lengths using the brute force online simulator are shown in Table 8. According to the use of brute force mechanism, this algorithm combines the alphabet and numeric to come up the set of character. So, the result of guessing 6 characters length and 7 characters length does spend more time to get the correct one, after they enter many sets of characters into the system. As presented in Table 12, there are 56,800, 235,584 combinations of 26 characters and 10 numeric; the time to break is at least 1 hour to guess the right answer. Like 6 characters length, there are 3,521,614,606,208 combinations which must use nearly 69 hours to break. Therefore, the proposed

Text-based CAPTCHA that consists of 6 or 7 characters has chances to be broken close to zero.

Table 10 The result of the brute force online simulator to guess 6 characters length and 7 characters length

Character length	Character combination	Time	Days
6	56,800, 235,584	1 hour and 10 minutes	0.05
7	3,521,614,606,208	68 hours and 33 minutes	2.85

4.4.2 Typing CAPTCHA Simulator Bot

For the experiment with the Emulator bot, the simulator reading a set of character of CAPTCHA and then input the characters. The results show that the attempt of this simulator bot cannot break this Text-based CAPTCHA because the system cannot capture pressing time of each characters.

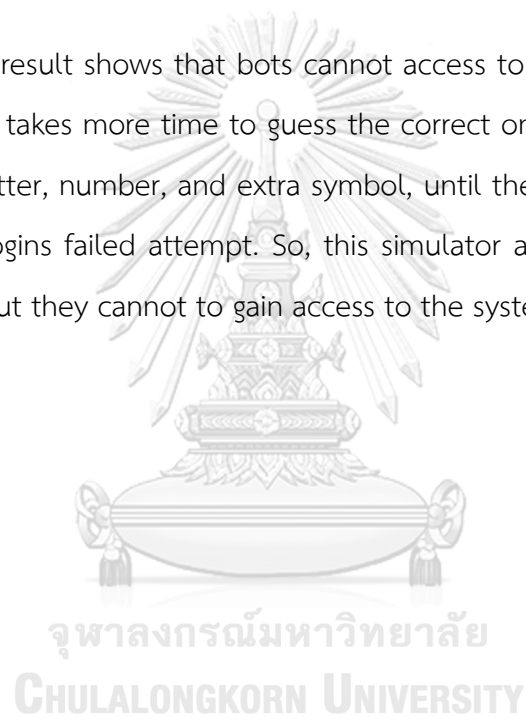
4.4.3 Typing CAPTCHA with Simulator Bot including special conditions

For the experiment with the Typing CAPTCHA simulator bot with special conditions, this simulator imitates the typing selected gender with time delay as similar to the human typing, and that is including with pressure and fingertip. The starting time delay is zero, and it is increasing every 1 milliseconds for every round until the bot can obtain the true value, then increasing pressure every 0.001 unit for every round, and the increasing fingertip every 1 unit for every round. So, table 11 shows the results that it can break the Text-based CAPTCHA, “r t h r o s” and “e t t o a t r”.

Table 11 The result of the simulator bot with special condition to crack 6 characters length and 7 characters length

Character length	Results
6	Cannot access
7	Cannot access

For Table 11, the result shows that bots cannot access to the system. However, the brute force attack takes more time to guess the correct one by trying every possible combination of letter, number, and extra symbol, until they get the correct one. So, there is several logins failed attempt. So, this simulator attempt to crack this Text-based CAPTCHA but they cannot to gain access to the system.



Chapter 5

Discussion

Biometrics is an important tool for security and privacy system in the past few years. New technological devices are always installed at least one security system using biometrics as a key for granting the service, such as fingerprint scan, face recognition, voice recognition, or retina scan. However, these biometrics can be degraded according to the user's age or some other external factors such as plastic surgery, and accident. Therefore, in such cases, the biometrics will not work as expected under uncontrolled circumstances. Thus, this research intends to propose undegradable factors for creating an authentication classification model so the system will be highly protection. From this aim, the supported factors are gender, the finger features, including the fingertip. Nevertheless, the values of the fingertip and the finger features, which are dwell time, latency time, flight time, interval time, and up-to-up time, must be obtained only when the user types the his/her personal information as full-name, surname, email, and phone number. According to the condition of typing user's full name, surname, email, and phone number, these are all data that any users are intimate as they frequently use in daily lives. As the fact that people must use their name and surname with email address and phone number for registering to join every activity or organization since they are young. Therefore, the typing patterns of these information of a person, or so called as the keystroke dynamic, must be unchanged even when they are getting older. So, the keystroke dynamic of each user will be unique and difficult to be emulated. Furthermore, the outcomes of the factor's suitability testing based on 4 different classification models have confirmed that using these factors can identify the right person with accuracy nearly 100%. Thus, the intruders cannot use the traditional attacking methods such as shoulder surfing, guessing, dictionary attacks and brute force attacks to gain access to the users' devices. Moreover, the proposed factors are

much reliable than other biometrics because some factors such as gender is usually unchanged by time. In addition, the values of all finger features and fingertip may be affected by time but it is still in the acceptable limit. Thus, when combining gender with the finger features and the fingertip, these classification factors will be much reliable than other biometrics. Table 14 shows the performance comparisons among authentication classification factors

Table 12 Performance comparison among authentication classification factors

Feature	Attack			Sustainability	Convenience	Efficiency
	Traditional	Bots	3 rd party			
Fingerprint [40]	/	/	/	/	/	99.00%
Iris [39]	/	/	/	/	/	99.00%
Retina [56]	/	/	/	/	/	97.50%
Teeth + image + voice [85]	/	/	/	/	/	96.00%
Keystroke [86]	/	/	/	/	/	97.40%
Proposed method	/	/	/	/	/	99.97%

Note: Proposed method is Gender + Dwell_time + Latency_time + Flight_time + Interval_time + Up_to_up_time + Finger pressure +Fingertip

Table shows the performance comparison among key factors that are used in the authentication classification models which were proposed by other researchers. This

comparison indicates that many researchers tried to look for highly efficient factors from combination among biometrics, but the most researchers did not focus on the degradation of some biometrics. Therefore, this research has focus on an additional factor that helps the authentication routine can maintain without worrying for any alters of users but still provides high accuracy in the authentication outcome. Moreover, all factors proposed in this research can be implemented in any touchscreen devices since it also implements a fingertip as an authentication indicator that can be obtained only when users use the touchscreen or the touchpad.

CAPTCHAs are the most popular method to distinguish bots from human. In addition, there are variety of CAPTCHAs that have been proposed. Unfortunately, a serious problem is the CAPTCHA farm where a third party hired people to automatically solve CAPTCHA via API. Thus, all type of CAPTCHAs can easily be solved. Therefore, this research applies the biometrics, such as keystroke dynamics, as a part of CAPTCHA solving to protect the problem of CAPTCHA farm and to authenticate the user at once.



This paper proposed a new type of Text-based CAPTCH that is derived based on the typing behavior of human. This proposed Text-based CAPTCHA can be used to identify the person because of different typing skills. The skill of typing of each person has difference, those skill rely on the word that frequency to type. In this case, profile information typing as full-name, surname, email, and phone number is high frequency type when register, or login into account in any applications. So, this research designs each Text-based CAPTCHA according to a set of characters that each user is well versed in typing. This design approach can be called as individual design

that the top ten of single characters and digraph characters of each person are chosen for randomly generated individual Text-based CAPTCHA.

To generate individual Text-based CAPTCHA, this proposed system generates Text-based CAPTCHAs based on individual dataset, that contains 15 characters: 10 single characters, and 5 digraph characters. As the Text-based CAPTCHA in this research has 6 digits, the sample space of all possible Text-based CAPTCHA contains 15^6 elements or 11,390,625 elements. So, the probability for an intruder to guess and achieve the 1 CAPTCHA attack is showed as:

$$\frac{1}{11,390,625} = 0.00000008779;$$

So, this probability is close to zero. Thus, it can conclude that the chance to break this Text-based CAPTCHA is hardly to occur.

Though CAPTCHA was implemented to distinguish between human and bots, the outcome of this research has changed the role of CAPTCHA to replace the traditional login system. This can occur because the individual Text-based CAPTCHAs are generated by individual datasets that related to individual users. The testing result shows that the accuracy to detect the right user is 100% while the imposter will be rejected as needed because of keystroke dynamics features' verification.

Moreover, after testing with the online bot attack under the brute force technique to guess Text-based CAPTCHA, this test shows times spent to hack the 6- or 7-digits CAPTCHA were longer than the time limit setting by the login system. Besides, the

test by the Time delay emulator bot has shown the same result as the online bot attack. Therefore, these two tests confirm that the proposed Text-based CAPTCHA is strong and hardly to be hacked by any available techniques.



Chapter 6

Conclusion

Text-based CAPTCHA are most widely used and deployed in major websites which distinguishes bots from actual human being. Many researchers designed very hard CAPTCHA to protect websites from efficient bots; however, that the clever bots cannot pass into the website as same as human. So, CAPTCHA should be friendly to human. Moreover, CAPTCHA business is introduced to solve CAPTCHA via API to automate solving CAPTCHA. Therefore, to prevent unauthorized as bots and CAPTCHA farm, a new form of CAPTCHA is introduced. This research proposed the new form of CAPTCHA that is built upon the user's profile and biometrics so as to gain the individual Text-based CAPTCHA that cannot be broken by bots or CAPTCHA farms. A new benchmark of Text-based CAPTCHA is adopted from the concept of the keystroke dynamics, including individual dataset that is obtained from personal data typing. For performance evaluation using Gradient Boosted Tree, the result shows 100% accuracy to identify the real user. So, the proposed Text-based CAPTCHA is not just a CAPTCHA, it can identify the user and be used as the password. Besides, the challenge of this new Text-based CAPTCHA is its user friendly as a simple Text-based CAPTCHA will be displayed to users when they are logging in.

Chapter 7

Future Work

The next step to be processed is to increase the strength of the Text-based CAPTCHA by adding the location of the user and some other personal information as part of the Text-based CAPTCHA to fully protect all types of threats. Besides, the invented Text-based CAPTCHA must be user friendly, easy to be read, and understandable for users.



Appendix

1. Statistics test for Gender's effect

- Dwell time

Test Statistics^a

	dwell
Mann-Whitney U	7063227.000
Wilcoxon W	15788980.00
Z	-11.435
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Interval time

Test Statistics^a

	interval
Mann-Whitney U	5681320.500
Wilcoxon W	12184741.50
Z	-12.978
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Latency time

Test Statistics^a

	latency
Mann-Whitney U	6438880.500
Wilcoxon W	13319075.50
Z	-8.552
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Flight time

Test Statistics^a

	flight
Mann-Whitney U	6282577.000
Wilcoxon W	13203637.00
Z	-10.674
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Up to up time

Test Statistics^a

	uptpup
Mann-Whitney U	6274016.000
Wilcoxon W	13198797.00
Z	-10.729
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Finger pressure

Test Statistics^a

	pressure
Mann-Whitney U	4960685.500
Wilcoxon W	11207030.50
Z	-15.615
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

- Fingertip

Test Statistics^a

	radius
Mann-Whitney U	7492505.500
Wilcoxon W	15343208.50
Z	-13.709
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: gender

2. Statistics test for Fingertip's effect

- Pressure

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of pressure is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

- Dwell time

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of dwell is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

- Interval time

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of interval is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

- Latency time

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of latency is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

- Flight time

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of flight is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

- Up to up

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of uptup is the same across categories of radius.	Independent-Samples Kruskal-Wallis Test	.000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is .05.

3. Factors Suitability Testing

- Accuracy

User	Deep learning	Decision Tree	Random Forest	Gradient Boosted Trees
1	94.2	94	94	94.75
2	99.67	99.79	99.98	94.16
3	95.5	95.5	91.7	95.5
4	92.8	92.8	92.3	92.3
5	93.1	92.8	92.8	92.8
6	93.5	93.5	93.5	93.5
7	93	91.5	91.7	92.1
8	92.2	92.5	72.2	93.3
9	97.5	97	93.6	96.9
10	99.3	99.3	99.3	99.3
11	94.4	94.2	75.6	95.4
12	92.6	91.6	91.8	92
13	90.3	92.6	92.9	92.5
14	91.8	91.8	91.8	91.8
15	90.4	95.3	95.3	95.3
16	96.8	96.6	96.7	96.6
average	94.191875	94.424375	91.57375	94.263125

- Precision

User	Deep learning	Decision Tree	Random Forest	Gradient Boosted Trees
1	94.51	94.62	94.7	95.91
2	96.94	98.03	99.72	99.02
3	97.25	95.9	95.83	96.93
4	96.2	92.56	92.6	94.88
5	97.21	92.91	92.9	97.19
6	96.04	93.54	93.5	95.29
7	96.14	91.66	92.29	95.94
8	97.44	94.13	93.61	96.99
9	98.61	98.28	98.09	98.56
10	99.53	99.63	99.56	99.72
11	96.59	94.8	94.74	96.97
12	97.14	91.84	94.42	97.05
13	96.83	93.72	93.52	96.86
14	96.59	91.75	91.75	96.06
15	96.96	95.85	95.6	96.34
16	98.52	97.43	97.48	98.49
average	97.03125	94.790625	95.019375	97.0125

- Recall

User	Deep learning	Decision Tree	Random Forest	Gradient Boosted Trees
1	94.51	99.68	99.8	98.61
2	99.72	99.82	99.97	99.97
3	91.36	99.59	99.71	96.98
4	77.85	99.75	99.7	89.24
5	87.25	99.8	99.97	91.01
6	86.32	99.75	99.9	94.2
7	93.8	99.72	99.54	93.11
8	94.22	98.79	99.75	95.12
9	99.26	99.31	99.56	98.87
10	99.65	99.81	99.95	99.72
11	93.05	99.7	99.83	97.4
12	95.76	99.92	99.09	96.96
13	86.92	99.02	99.4	90.77
14	81.77	99.75	99.82	88.47
15	94.19	99.32	99.68	95.58
16	97.52	99.57	99.74	97.5
average	92.071875	99.58125	99.713125	95.219375

- Execute time

User	Deep learning	Decision Tree	Random Forest	Gradient Boosted Trees
1	25	11	35	47
2	25	11	119	32
3	24	11	34	28
4	24	11	31	27
5	24	11	80	34
6	25	11	31	28
7	29	11	34	30
8	26	12	188	32
9	24	11	84	35
10	25	13	31	35
11	26	11	154	35
12	24	11	149	41
13	24	11	15	34
14	25	13	31	29
15	25	11	23	33
16	25	11	123	58
average	25.3125	11.3125	72.625	34.875

REFERENCES

1. Ahn, L.v., M. Blum, and J. Langford, *Telling humans and computers apart automatically*. Commun. ACM, 2004. **47**(2): p. 56–60.
2. Datawow. *Solving text-based CAPTCHA with Machine Learning — part 1*. 2020 [cited 2020 10 November]; Available from: <https://datawow.io/blogs/solving-text-based-captcha-part1>.
3. Moradi, M. and M. Keyvanpour, *CAPTCHA and its Alternatives: A Review*. Security and Communication Networks, 2015. **8**(12): p. 2135-2156.
4. Baird, H.S., A.L. Coates, and R.J. Fateman, *PessimalPrint: a reverse Turing test*. International Journal on Document Analysis and Recognition, 2003. **5**(2): p. 158-163.
5. Egele, M., et al., *CAPTCHA smuggling: hijacking web browsing sessions to create CAPTCHA farms*, in *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010, Association for Computing Machinery: Sierre, Switzerland. p. 1865–1870.
6. Brownlee, J. *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*. 2020 [cited 2020 10 November]; Available from: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/?fbclid=IwAR2u7y-O4xl_fgKOr4ILgbM-Mb9kurrer7op09pgHdV5tMfS45iEY-7ZGOE.
7. Baird, H.S. and K. Popat, *Human Interactive Proofs and Document Image Analysis*, in *Proceedings of the 5th International Workshop on Document Analysis Systems V*. 2002, Springer-Verlag. p. 507–518.
8. Ahn, L.V., et al., *CAPTCHA: using hard AI problems for security*, in *Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques*. 2003, Springer-Verlag: Warsaw, Poland. p. 294–311.
9. von Ahn, L., et al., *reCAPTCHA: human-based character recognition via Web security measures*. Science, 2008. **321**(5895): p. 1465-8.
10. Thangavelu, S. and T. Purusothaman, *Analysis of Different Text Based Captcha*

- Methods*. International Journal of Computer Science and Mobile Computing (IJCSMC), 2015. **4**: p. 128-133.
11. Alqahtani, F.H. and F.A. Alsulaiman, *Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study*. Computers & Security, 2020. **88**: p. 101635.
 12. Kwon, S. and S. Cha, *CAPTCHA-based image annotation*. Information Processing Letters, 2017. **128**: p. 27-31.
 13. Gossweiler, R., M. Kamvar, and S. Baluja, *What's up CAPTCHA?: a CAPTCHA based on image orientation*. 2009. 841-850.
 14. Soupionis, Y. and D. Gritzalis, *Audio CAPTCHA: Existing solutions assessment and a new implementation for VoIP telephony*. Computers & Security, 2010. **29**(5): p. 603-618.
 15. Bigham, J. and A. Cavender, *Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use*. 2009. 1829-1838.
 16. Aldwairi, M., S. Mohammed, and M.L. Padmanabhan, *Efficient and secure flash-based gaming CAPTCHA*. Journal of Parallel and Distributed Computing, 2020. **142**: p. 27-35.
 17. Rao, K., K. Sri, and G. Sai, *A Novel Video CAPTCHA Technique To Prevent BOT Attacks*. Procedia Computer Science, 2016. **85**: p. 236-240.
 18. Wang, X. and M. Reiter, *A multi-layer framework for puzzle-based denial-of-service defense*. Int. J. Inf. Sec., 2008. **7**: p. 243-263.
 19. Ali, F. and F. Karim, *Development of CAPTCHA system based on puzzle*. 2014. 426-428.
 20. Konheim, A., *Automated teller machines: their history and authentication protocols*. Journal of Cryptographic Engineering, 2016. **6**.
 21. Markert, P., et al., *This PIN Can Be Easily Guessed: Analyzing the Security of Smartphone Unlock PINs*. 2020. 286-303.
 22. Wang, D., Q. Gu, and P. Wang, *Understanding Human-Chosen PINs: Characteristics, Distribution and Security*. 2017.
 23. Roth, V., K. Richter, and R. Freidinger, *A PIN-entry method resilient against shoulder surfing*. 2004. 236-245.

24. Lee, M.-K., *Security Notions and Advanced Method for Human Shoulder-Surfing Resistant PIN-Entry*. Information Forensics and Security, IEEE Transactions on, 2014. **9**: p. 695-708.
25. Nyang, D., et al., *Two-Thumbs-Up: Physical Protection for PIN Entry Secure against Recording Attacks*. Computers & Security, 2018. **78**.
26. KrebsonSecurity. *Secret Service Warns of Surge in ATM 'Wiretapping' Attacks*. 2018 [cited 2020 4 October]; Available from: <https://krebsonsecurity.com/2018/09/secret-service-warns-of-surge-in-atm-wiretapping-attacks/>.
27. Paganini, P. *ATM Attacks Are Skyrocketing*. 2018 [cited 2020 9 October]; Available from: <https://resources.infosecinstitute.com/topic/atm-attacks-are-skyrocketing/#gref>.
28. Business, B.s. *What Were the Most Common Passwords in 2019?* 2019 [cited 2020 20 October].
29. MASTROIANNI, B. *These were the 25 worst passwords of 2015*. 2016 [cited 2020 20 October]; Available from: <https://www.cbsnews.com/news/these-were-the-25-worst-passwords-of-2015/>.
30. Song, J., et al., *Alphapwd: A Password Generation Strategy Based on Mnemonic Shape*. IEEE Access, 2019. **7**: p. 119052-119059.
31. Conklin, W., G. Dietrich, and D. Walz, *Password-Based Authentication: A System Perspective*. Vol. 37. 2004.
32. Alomari, R. and J. Thorpe, *On password behaviours and attitudes in different populations*. Journal of Information Security and Applications, 2019. **45**: p. 79-89.
33. Grassi, P.A., et al., *Digital identity guidelines: Authentication and lifecycle management [including updates as of 12-01-2017]*. 2017.
34. Li, Q., P. Dong, and J. Zheng, *Enhancing the Security of Pattern Unlock with Surface EMG-Based Biometrics*. Applied Sciences, 2020. **10**(2): p. 541.
35. Cha, S., et al. *Boosting the guessing attack performance on Android lock patterns with smudge attacks*. in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 2017.

36. Von Zezschwitz, E., et al. *Easy to draw, but hard to trace? On the observability of grid-based (un) lock patterns*. in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015.
37. Giot, R., M. El-Abed, and C. Rosenberger, *Fast computation of the performance evaluation of biometric systems: Application to multibiometrics*. *Future Generation Computer Systems*, 2013. **29**(3): p. 788-799.
38. Pankanti, S., S. Prabhakar, and A.K. Jain, *On the individuality of fingerprints*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. **24**(8): p. 1010-1025.
39. Bulman, P. *NIST Study Shows Computerized Fingerprint Matching Is Highly Accurate*. 2004 [cited 2020 7 September]; Available from: <https://www.nist.gov/news-events/news/2004/07/nist-study-shows-computerized-fingerprint-matching-highly-accurate>.
40. Jain, A.K., *Biometric recognition*. *Nature*, 2007. **449**(7158): p. 38-40.
41. Baguley, R. *Inside The Apple iPhone 5s Fingerprint Sensor*. 2013 [cited 2020 8 September]; Available from: <https://medium.com/people-gadgets/inside-the-apple-iphone-5s-fingerprint-sensor-b2e80b75c22>.
42. Jeffery, L. *Biometrics and the future of payment transactions*. 2019 [cited 2020 8 September]; Available from: <https://www.biometricupdate.com/201912/biometrics-and-the-future-of-payment-transactions>.
43. Chawla, P.M. and P. Mishra, *A Fast Biometric Fingerprint Payment System*. *Journal of Digital Integrated Circuits in Electrical Devices*, 2017. **2**(3): p. 1-11.
44. Ulery, B., et al., *Accuracy and reliability of forensic latent fingerprint decisions*. *Proceedings of the National Academy of Sciences of the United States of America*, 2011. **108**: p. 7733-8.
45. Feng, D., P. Wang, and L. Zu. *Design of Attendance Checking Management System for College Classroom Students Based on Fingerprint Recognition*. in *2020 Chinese Control And Decision Conference (CCDC)*. 2020.
46. Alsunaidi, S.J. and A.M. Almuhaideb, *Investigation of the optimal method for generating and verifying the Smartphone's fingerprint: A review*. *Journal of King*

- Saud University - Computer and Information Sciences, 2020.
47. Jain, A., K. Nandakumar, and A. Nagar, *Biometric Template Security*. EURASIP Journal on Advances in Signal Processing, 2008. **2008**.
 48. Yang, W., et al., *Security and Accuracy of Fingerprint-Based Biometrics: A Review*. Symmetry, 2019. **11**: p. 141.
 49. Drahansky, M., et al., *Influence of Skin Diseases on Fingerprint Recognition*. Journal of biomedicine & biotechnology, 2012. **2012**: p. 626148.
 50. Jo, Y.-H., et al., *Security Analysis and Improvement of Fingerprint Authentication for Smartphones*. Mobile Information Systems, 2016. **2016**: p. 1-11.
 51. Goodin, D. *Attackers can bypass fingerprint authentication with an ~80% success rate*. 2020 [cited 2020 4 September]; Available from: <https://arstechnica.com/information-technology/2020/04/attackers-can-bypass-fingerprint-authentication-with-an-80-success-rate/?fbclid=IwAR0OdeOVtrD46T8ipSjLx9lgPU5XzYhOEWjup5Bl3wPZafzfgAGx9L-5Flo>.
 52. Brown, C.S. *Hackers able to unlock smartphones by lifting fingerprints off a drinking glass*. 2019 [cited 5 September; Available from: <https://www.androidauthority.com/smartphone-fingerprint-hackers-1049088/>.
 53. Whittaker, Z. *Hackers can remotely steal fingerprints from Android phones*. 2015 [cited 2020 5 September]; Available from: <https://www.zdnet.com/article/hackers-can-remotely-steal-fingerprints-from-android-phones/>.
 54. Lee, P. and H. Ewe, *Individual Recognition Based on Human Iris Using Fractal Dimension Approach*. 2004. 467-474.
 55. Barkhoda, W., et al., *Retina identification based on the pattern of blood vessels using fuzzy logic*. EURASIP Journal on Advances in Signal Processing, 2011. **2011**.
 56. Sadikoglu, F. and S. Uzelaltinbulat, *Biometric Retina Identification Based on Neural Network*. Procedia Computer Science, 2016. **102**: p. 26-33.
 57. King, R. *Explainer: Retinal Scan Technology*. 2020 [cited 2020 9 September]; Available from: <https://www.biometricupdate.com/201307/explainer-retinal-scan-technology>.

58. Frucci, M., et al., *Severe: Segmenting vessels in retina images*. Pattern Recognition Letters, 2016. **82**: p. 162-169.
59. Bell, G., *Strengthening CAPTCHA-based Web security*. First Monday J., 2012. **17**.
60. Kolupaev, A. and J. Ogjenko, *CAPTCHAs: Humans vs. Bots*. IEEE Security & Privacy, 2008. **6**(1): p. 68-70.
61. Yan, J. and A.S.E. Ahmad. *Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms*. in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. 2007.
62. Song, C. and V. Shmatikov, *Fooling OCR Systems with Adversarial Text Images*. 2018.
63. anti-captcha.com. *CAPTCHA Solving Service*. 2020 [cited 2020 1 October]; Available from: <https://anti-captcha.com/mainpage>.
64. Motoyama, M., et al., *Dirty Jobs: The Role of Freelance Labor in Web Service Abuse*. 2011. 14-14.
65. Dzieza, J. *WHY CAPTCHAS HAVE GOTTEN SO DIFFICULT*. 2019 [cited 2020 1 October]; Available from: <https://www.theverge.com/2019/2/1/18205610/google-captcha-ai-robot-human-difficult-artificial-intelligence>.
66. Netacea. *PART TWO: WHAT ARE CAPTCHA FARMS?* 2019 [cited 1 October; Available from: <https://www.netacea.com/blog/what-are-captcha-farms/>.
67. Truong, H., C. Turner, and C. Zou, *iCAPTCHA: the next generation of CAPTCHA designed to defend against 3rd party human attacks*. 2011. 1-6.
68. Wei, T., A.B. Jeng, and H. Lee. *GeoCAPTCHA — A novel personalized CAPTCHA using geographic concept to defend against 3rd Party Human Attack*. in *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*. 2012.
69. Ye, Q., et al. *DDIM-CAPTCHA: A Novel Drag-n-Drop Interactive Masking CAPTCHA against the Third Party Human Attacks*. in *2013 Conference on Technologies and Applications of Artificial Intelligence*. 2013.
70. Li, Z. and Q. Liao. *CAPTCHA: Machine or Human Solvers? A Game-Theoretical Analysis*. in *2018 5th IEEE International Conference on Cyber Security and*

- Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. 2018.
71. Bera, A., D. Bhattacharjee, and M. Nasipuri, *Hand Biometric Verification with Hand Image-Based CAPTCHA*. 2018. p. 3-18.
 72. Alreshoodi, L.A. and S.A. Alsuhibany, *A Proposed Methodology for Detecting Human Attacks on Text-based CAPTCHAs*.
 73. Monrose, F. and A.D. Rubin, *Keystroke dynamics as a biometric for authentication*. *Future Generation Computer Systems*, 2000. **16**(4): p. 351-359.
 74. Monrose, F. and A. Rubin, *Authentication via keystroke dynamics*, in *Proceedings of the 4th ACM conference on Computer and communications security*. 1997, Association for Computing Machinery: Zurich, Switzerland. p. 48-56.
 75. Hempstalk, K., E. Frank, and I. Witten, *One-Class Classification by Combining Density and Class Probability Estimation*. 2008.
 76. Cho, S., et al., *Web-Based Keystroke Dynamics Identity Verification Using Neural Network*. *Journal of Organizational Computing and Electronic Commerce*, 2000. **10**: p. 295-307.
 77. Villani, M., et al., *Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions*. 2006. 39-39.
 78. Bergadano, F., D. Gunetti, and C. Picardi, *User authentication through keystroke dynamics*. *ACM Trans. Inf. Syst. Secur.*, 2002. **5**(4): p. 367-397.
 79. Peacock, A., K. Xian, and M. Wilkerson, *Typing patterns: a key to user identification*. *IEEE Security & Privacy*, 2004. **2**(5): p. 40-47.
 80. Saevanee, H. and P. Bhattarakosol. *Authenticating User Using Keystroke Dynamics and Finger Pressure*. in *2009 6th IEEE Consumer Communications and Networking Conference*. 2009.
 81. Trojahn, M., F. Arndt, and F. Ortmeier, *Authentication with Time Features for Keystroke Dynamics on Touchscreens*. 2013. 197-199.
 82. Draffin, B., J. Zhu, and J. Zhang, *KeySens: Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction*. Vol. 130. 2014. 184-201.
 83. Coakley, M.J., J.V. Monaco, and C.C. Tappert. *Keystroke biometric studies with short numeric input on smartphones*. in *2016 IEEE 8th International Conference*

- on *Biometrics Theory, Applications and Systems (BTAS)*. 2016.
84. Flores, V. and B. Keith Norambuena, *Gradient Boosted Trees Predictive Models for Surface Roughness in High-Speed Milling in the Steel and Aluminum Metalworking Industry*. Complexity, 2019. **2019**: p. 1-15.
85. Kim, D. and K. Hong, *Multimodal biometric authentication using teeth image and voice in mobile environment*. IEEE Transactions on Consumer Electronics, 2008. **54**(4): p. 1790-1797.
86. Krishnamoorthy, S., et al., *Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning*. 2018. 50-57.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

VITA

NAME Nilobon Nanglae

DATE OF BIRTH 19 February 1983

PLACE OF BIRTH Chiangrai

INSTITUTIONS ATTENDED -Master's Degree of Computer Science, Chulalongkorn University, Thailand
Thesis: Authentication Indicators Using Bio-Detection Function with Text-Based CAPTCHA / Nilobon Nanglae
-Bachelor's Degree of Liberal Arts, Mae Fah Luang University, Thailand

HOME ADDRESS 235 Moo.19 Phokhon Road, Soi 3, Tambon Wiang, Amphoe Muang Chiang Rai, Chiang Rai, 57000, Thailand

PUBLICATION -N. Nanglae and P. Bhattarakosol, "Attitudes towards Text-based CAPTCHA from developing countries," 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, 2015, pp. 1-4, doi: 10.1109/ECTICon.2015.7207116.
-NANGLAE, Nilobon; BHATTARAKOSOL, Pattarasinee. Authentication Indicators Based Bio-Detection Function with Text-based CAPTCHA. International Journal of Digital Content Technology and its Applications, 2014, 8.1: 10.
-N. Nanglae and P. Bhattarakosol, "A Study of Human Bio-detection Function under Text-Based CAPTCHA System," 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, Shanghai, 2012, pp. 139-144, doi: 10.1109/ICIS.2012.19.